POLITECNICO
MILANO 1863

POLITECNICO DI MILANO
MECHANICAL ENGINEERING
METHODS AND TOOLS FOR PRODUCT DESIGN

# ON OBJECT RECOGNITION FOR INDUSTRIAL AUGMENTED REALITY

Doctoral Dissertation of:
**Juan Carlos Arbeláez**

Supervisors:
**Prof. Roberto Viganò**
**Prof. Gilberto Osorio**

Tutor:
**Prof. Federico Cheli**

The Chair of the Doctoral Program:
**Prof. Daniele Rocchi**

30 Cycle

# Abstract

Like natural systems, man-made systems evolve to become more complex over time. Some reasons are market pressure, an increase of functionality, and adaptability to an already complex environment, among others.

Therefore, workers face fast-changing and challenging tasks along with all the product lifecycle that reach the human cognitive limits. Although nowadays some operations are automated, many of them still need to be carried out by humans because of their complexity.

In addition to management strategies and design for X, Industrial Augmented Reality (IAR) has proven to potentially benefit activities such as maintenance, assembly, manufacturing, and repair, among others. It is also supposed to upgrade the manufacturing processes by improving it, simplifying decision-making activities, reducing time and user movements, diminishing errors, and decreasing mental and physical effort.

Nevertheless, IAR has not succeeded in breaking out of the laboratories and establishing itself as a strong solution in the industry, mainly because technical and interaction components are far from ideal. Its advance is limited by its enabling technologies. One of its biggest challenges are the methods for understanding the surroundings considering the different domain variables that affect IAR implementations.

Thus, inspired by some systematical methodologies proposing that, for any problem-solving activity, it is required to define the characteristics that constrain the problem and the needs to be satisfied, a general frame of IAR was proposed through the identification of Domain Variables (DV), that are relevant characteristics of the industrial process in the previous Augmented Reality (AR) applications. These DV regard the user, parts, environment, and task that have an impact on the technical implementation and user performance and perception (Chapter 2).

Subsequently, a detailed analysis of the influence of the DV on technical implementations related to the processes intended to understand the surroundings was performed. The results of this analysis suggest that the DV influence the technical process in two ways. The first one is that they define the *boundaries* in the characteristics of the technology, and the second one is that they cause some *issues* in the process of understanding the surroundings (Chapter 3).

Further, an automatic method for creating synthetic datasets using solely the 3D model of the parts was proposed. It is hypothesized that the proposed variables are the main source of visual variations of an object in this context. Thus, the proposed method is derived from physically recreated light-matter interactions of this relevant variables. This method is aimed to create fully labeled datasets for training and testing surrounding understanding algorithms (Chapter 4).

Finally, the proposed method is evaluated in a study case of object classification of two cases: a particular industrial case, and a general classification problem (using classes of *ImageNet*). Results suggest that fine-tuning models with the proposed method reach comparable performance (no statistical difference) than models trained with photos. These results validate the proposed method as a viable alternative for training surrounding understanding algorithms applied to industrial cases (Chapter 5).

# Contents

# Acronyms

| | |
|---|---|
| **Symbols** | |
| 2D | Two-dimensional space. 31, 42, 58, 63, 81 |
| 3D | Three-dimensional space. I, 7, 31, 33, 40, 42, 45, 49, 57, 58, 60–64, 81, 82, 84, 86, 95, 96, 99, 102, 113–115, 133, 139 |
| **A** | |
| AHP | Analytic Hierarchy Process. 14 |
| AI | Artificial Intelligence. 40 |
| AP | Average Precision. 47 |
| AR | Augmented Reality. I, III, 1–6, 8, 9, 11–15, 23–25, 27–38, 42, 45–47, 58, 130, 133, 134, 137, 143 |
| **B** | |
| BB | Bounding Box. 99 |
| BRDF | Bidirectional Reflectance Distribution Function. 68, 69, 73–75, 78, 86, 87 |
| **C** | |
| CNN | Convolutional Neural Network. V, 8, 60, 62–64, 84, 107–110, 112–114, 116, 118, 120, 122, 124, 126, 128, 130–132, 140 |
| CPU | Central Processing Unit. 81 |
| CRF | Conditional Random Field. 49–51, 55, 144 |
| **D** | |
| DA | Domain Adaptation. V, 8, 57–59, 103, 107, 108, 130 |
| DAE | Digital Asset Exchange. 99, 115 |
| DNN | Deep Neural Network. 59 |

| | |
|---|---|
| DV | Domain Variables. I, III, IV, 9, 11, 12, 14–17, 19, 21, 24–28, 31–37, 40–43, 45, 47–55, 57, 60, 79–82, 85, 104, 114, 134, 137, 139, 143, 144 |
| **E** | |
| EM | Electromagnetic. 6 |
| **G** | |
| GPS | Global Positioning System. 5, 31 |
| GPU | graphics Processing Unit. 81, 84 |
| **H** | |
| HCI | Human Computer Interaction. 2 |
| HDR | High Dynamic Range. 78, 97 |
| HMD | Head Mounted Displays. 29, 30, 34 |
| **I** | |
| IAR | Industrial Augmented Reality. I, III, 1, 3, 9, 11–14, 28, 34, 35, 51–55, 58, 79, 107, 109, 130, 131, 133, 134, 143, 144 |
| IBL | Image-Based Lighting. 78, 97 |
| IOR | Index Of Refraction. 60, 64–66, 69, 80, 83, 86, 88–90, 92–95, 98, 138–140 |
| IOT | Internet Of Things. 1, 31 |
| IR | Infrared Radiation. 42, 43 |
| ITS | Intelligent Tutoring Systems. 33 |
| **J** | |
| JSON | JavaScript Object Notation. 115 |
| **L** | |
| LED | light Emitting Diode. 45 |
| **M** | |
| MCMC | Markov chain Monte Carlo. 49–51, 55, 144 |
| ML | Machine Learning. 40, 59, 108, 113, 133, 135 |
| **N** | |
| NDF | Normal Distribution Function. 75, 76 |
| NN | Neural Network. 47, 49–52, 63, 90, 102, 109, 135, 144 |
| NUI | Natural User Interface. 34 |
| **O** | |

| | |
|---|---|
| OR | Object Recognition. III, IV, 5–8, 27, 30, 40, 46–49, 51, 53, 55, 58, 60–63, 137, 143 |
| ORB | Oriented FAST and Rotated BRIEF. 8, 47 |
| **P** | |
| PBS | Physically Based Shading. IV, 9, 57, 58, 60–80, 82, 84–86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 107, 108, 115, 128, 130, 134, 139 |
| PCA | Principal Component Analysis. 63 |
| PDF | Probability Density Function. 81–83, 102, 104, 114, 144 |
| POV | Point Of View. 20, 36, 43, 44, 48–51, 58–60, 63, 64, 68, 79, 81, 82, 103, 104, 114, 128 |
| **R** | |
| RFID | Radio-frequency identification. 5, 28, 29 |
| **S** | |
| SIFT | Scale Invariant Feature Transform. 8, 31, 47, 49, 53 |
| SURF | Speeded-Up Robust Features. 49 |
| SVM | Support Vector Machine. 49, 51, 55, 62, 64, 144 |
| **T** | |
| TL | Transfer Learning. 59, 60, 108 |
| TS | Technical Systems. 12 |
| **U** | |
| UI | User Interface. 4 |
| **V** | |
| VR | Virtual Reality. 2 |

# CHAPTER 1

# Introduction

Products are an important part of our daily life, they help us to survive, improve our life quality and even satisfy our more complex human needs [119]. These products (Engineered, discrete, physical [221]) like natural systems evolve over time to become more complex. Some of the reasons are due to market pressure, the increase of functionality, and adaptability to an already complex environment, among others [15, 145].

Nowadays, some of the trends that drive such increase of complexity in products are mass customization [28], Internet Of Things (IOT) [225] and multi-functional products [53]. Companies are required to create innovative products in a highly competitive and dynamic environment within short periods. Low costs and high degrees of variation in small productions are also expected [170].

Therefore, workers face fast-changing and challenging tasks along with all the product lifecycle that reach the human cognitive limits [18]. And, although nowadays some operations are automated, many of them still need to be carried out by humans because of their complexity. Usually, these tasks include many actions in which acquiring the required dexterity could take a lifetime of experience and practice [242].

In addition to management strategies and design for X [65], a set of promising technologies covered under the name of Digital Manufacturing, are aimed to address products complexity, increase products quality and reduce production times and cost. The core idea of this set of technologies is to close the gap between the products definition and their actual implementation [41].

Among this set of technologies, we found Industrial Augmented Reality (IAR), that is related to the use of Augmented Reality (AR) as support of industrial field activities.

## 1.1 Augmented Reality

If reality is everything that exist, how is possible to be augmented? that is because is not the reality that is being augmented, is the perception of reality the one that is augmented [107].

Further, Augmented Reality (AR) allows the user to perceive the real world, with virtual objects superimposed or mixed with the real world [19, 21]. And can be seen as a set of innovative Human Computer Interaction (HCI) techniques [232] that enriches the way of experience the real world by inserting and interacting with virtual elements [255].

More formally and also one of the most accepted definitions is the one presented by Azuma in 1997 where AR meets three requirements [21]:

1. Combines real and virtual
2. Is interactive in real time
3. Is registered in three dimensions, meaning, the virtual elements need to be aligned to simulate the visual transformation between the camera and the object

In essence, the main goal of AR is to achieve a realistic blend of the virtual elements to the reality. Although this is an important characteristic for some applications, there are some other cases, like in the industrial field [74], where depending on the case it is not as important [186] and should not be over (or forced to) the effectiveness and usability.

It is notable that although the majority applications found in the review are intended for visual perception, AR is applicable for all the human senses and both adding (virtual elements) and removing (real elements) are valid by its definition [21, 37].

This contradicts especially the third item in the Azuma AR definition. Where, the AR that display information without being registered in the space is called Augmented Reality without context.The information is displayed similar to an electronic manual but reacting to context (context aware) avoiding to the user to change his or her attention to the task [166].

In a global frame, AR, can be described as a *Mixed Reality* that lies in the middle of a *Virtual Continuum* between a totally virtual environment (VR) and the real environment [156] as it is shown in Figure 1.1.

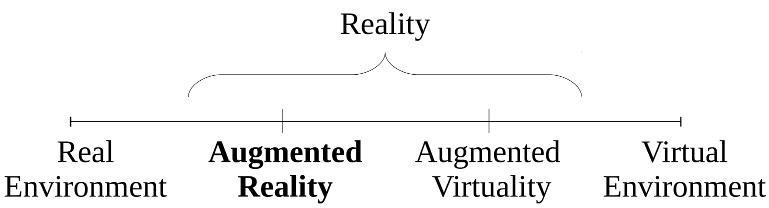


**Figure 1.1:** *Virtual Continuum of Milgram (1994). Modified from [156].*

Over the time, there has developed a more user-oriented definition, towards the interaction and task compelling, by adding a purpose to the definition, such as, simplifying the user's life by bringing virtual information [37, 144], assist the user in performing a task in a physical setting [63, 64, 200, 243], or simply, to facilitate human-computer interaction [89, 148].

Thus, currently, one of the most important issues regarding the implementation of AR is about the perceived usefulness and the ease of use that are the basis for the acceptance and adoption of any new technology [51]

This looser definition allows the development to be driven by the usability in terms of what type of media suits better in different situations. Which is especially convenient in industrial applications, where has been historically one of the major research fields. In which performance improvement and long times of use are usually required.

## 1.2 Industrial Augmented Reality

AR dates from the 90s with the development of a prototype that used a see-through head-mounted display, and combined with a tracking and registration system, it superimposed virtual information over the real world. This technology was aimed to support some wiring activities at Boeing and had shown potential benefits in terms of efficiency and costs reduction [38].

The application of AR in order to support some industrial processes is named Industrial Augmented Reality (IAR) [74]. And over the past decades, IAR has proven to potentially benefit activities such as maintenance, assembly, manufacturing, and repair, among others [74, 101, 133, 170, 200]. Where the main concept is to present in a natural way, real-time and context-aware information to the user to support the completion of some industrial task.

It is also supposed to upgrade the manufacturing processes by improving it [200], simplifying decision-making activities [76], reducing time and user's movements [98], diminishing errors [22] and decreasing mental and physical effort [213].

Similarly, even in fully automated production, AR is proposed to meet the need of a simple reacting interface for robots [122]. Further, AR is thought to be the future of human-computer interaction.

Nevertheless, IAR has not succeeded in breaking out of the laboratories and establishing itself as a strong solution in the industry, mainly because technical and interaction components are far from ideal [74]. This means that the use of AR still presents some disadvantages such as stress produced by its long-term usage [218], limits its usage to certain conditions or controlled environments [232], or requires a large development where their costs, are greater than the perceived benefits [74].

## 1.3 Challenges of Augmented Reality

Azuma proposed three main categories of obstacles for the spreading of AR [19], that can be though as a layered system that can describe the global behavior of AR (Figure 1.2). Where the user interaction is the relationship between the human and the technology.

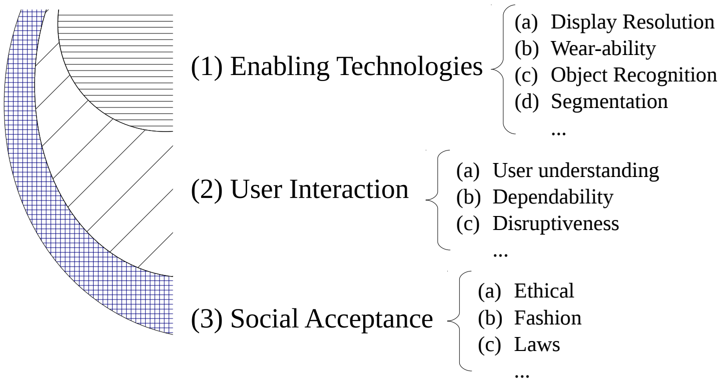


**Figure 1.2:** *General challenges of AR technology. Based on mayor AR obstacles from [19].*

Hence, there are characteristics of technology and how they affect the perceptions of the users, altogether framed in a set of social rules.

### 1.3.1 Social Acceptance

Once AR becomes part of the user's everyday life, is required to have a better understanding of how AR can affect the human values, in a psychological, behavioral point of view [200].

Further, the likability of the technology plays an important role in the expansion of a new technology. For instance, Kipper et al. exemplified this situation comparing the probability of success of new technology in countries such as Japan that have a culture of new technology adoption compared with others such as the United States or Europe. Where other factors as privacy and physical safety are required to play a major role in its future expansion [124].

### 1.3.2 User Interaction

The user interaction is related to the understanding of the human interfaces with the virtual elements, in terms of what and how the information should be displayed. Where it is supposed to achieve natural interactions, but technical limitations sometimes become a barrier [200]. Even though by its nature, AR presents a low learning curve, some other challenges emerge regarding the usability and how AR should stand by its efficiency compared to other solutions [124].

Another issue in the task supporting role of AR, is the overload and the over-reliance, meaning that the user interface should not overload the user. And at the same time, it should prevent the user from over-rely in the technology, avoiding important clues in the environment. For instance in driving assistance applications [223].

Thus the benefits of AR can be achieved when the User Interface (UI) is able to maximize the relevance and minimize the confusion of the virtual information regarding the real world [170]. Further it requires being situation aware related to the adaptation of the system to unplanned circumstances [133].

### 1.3.3 Enabling Technologies

These are the set of technologies required to develop an AR environment and are related with software and hardware [19]. A typically AR architecture, is composed by 6 modules [232]:

1. Video capture: live video stream
2. Image analysis and processing: computer vision algorithms for image processing
3. Tracking process: relative position of objects regarding the camera
4. Interaction Handling: human-computer interactions
5. Information management: retrieving information from different sources
6. Rendering kernel: visual representation of the data

Regarding the main definition of AR, the goal of blending the virtual and real elements in a non-distinguishable way has been achieved. But still, some technical issues such as dealing with occlusion, delays reduction in rendering process [200], and "bulky" hardware with low resolution do not provide a comfortable use, which compromises a possible industrial implementation [101].

Also, sensor accuracy in mobile systems such as optical sensors, accelerometers, GPS, gyroscopes, RFID among others, that are used to acquire the context information, is critical in applications where high level of accuracy is required, such as medical and some manufacturing processes [124, 164].

Additionally, one of the biggest challenges are the techniques that allow the system to understand the real world in unprepared environments. And, they are the key technical limitation that holds AR from becoming a new form of media [20].

Further, Object Recognition (OR) and the estimation of real objects state can be considered the core of AR because they are the processes that allow to link the real world elements with their virtual counterpart and define the actual conditions of the objects in real life.

## 1.4 Understanding the Real World - Object Recognition

Many different processes are required to achieve an understanding of the real world. Some of the basis processes for performing this task are related with: segmentation [252], tracking [16], registration [57] and Object Recognition (OR) (Section 3).

Therefore, one of the basic functions that need to be performed by an AR system is to recognize the interest objects. In this research we will focus on the Object Recognition (OR) as the process related to examining one or more images to evaluate which objects are present by using (usually) some knowledge about the appearance of the object (previously created model) [217].

At the beginning of this technology, pure hardware implementation was used (such as gyroscopes, accelerometers and ultrasonic) to try to solve OR and registration problem for AR. But, this can result very difficult to achieve since there is no feedback about how close or far the augmentation is from the real world [21]. However visual-based techniques will allow having a "closed-loop" in the tracking system in order to correct some registration problems and enforce the virtual and real world matching [21].

As consequence two main approaches have been developed over the time: visual and non-visual based (Figure 1.3). An example of non-visual-based is proposed by Laput et al. [134] where the Electromagnetic (EM) field noise, emitted by electrical and electromechanical objects during its operation, is used. This noise is transmitted to the user when he or she touches the object thanks to the human body conductivity. Then, it is possible by using a smartwatch to read and process this signal.

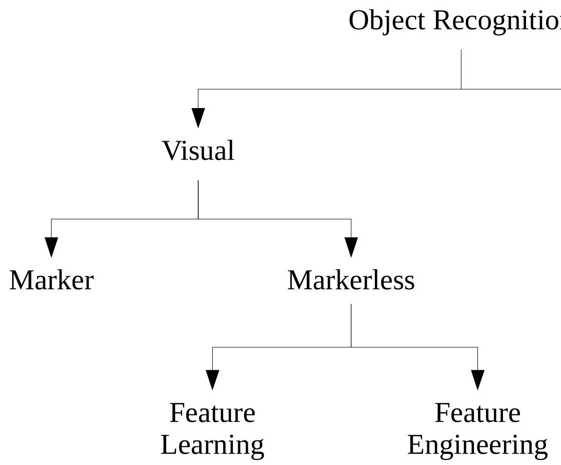


**Figure 1.3:** *General map of Object Recognition for Augmented Reality*

Some of the benefits of this kind of approach are that is not obstructive, it does not require instrumentation of the environment and it is somehow robust on-touch object detection. Nevertheless, many problems arrive with this kind of approach such as it is placed dependent, and is limited to objects that generate EM signals. Additionally, it is not possible to perform a registered augmentation [134].

In order to ease this difficult task, some applications proposed the use of *fiducials* or *fiducials markers* (Marker based on Figure 1.3). They are objects placed in the field of view of an optical system to be used as a reference. For example, they can be LEDs or other special elements such as flat pattern that is called marker. These markers were one of the most popular used systems, in order to ease the object recognition and the registration.

Nowadays, the existence of some frameworks (Metaio®, Vuforia® of Artoolkit®) allows the user to focus on developing the application content and logic but the capability of the systems to recognize objects in the real world is limited when there is not possible to use *fiducial markers* [133].

Where the general process can be described as it is proposed by Andreopolus (Figure

1.4) [13], the recognition is based on the difference between the features of the query image (input) and the features stored in a database.

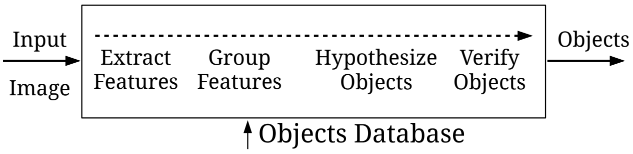


**Figure 1.4:** *Components of the recognition process. Adapted from [13]*

As consequence, the recognition system relays on the *natural features* which are derived from intrinsic characteristics of the object such as color distribution (texture), edges, shape, corners. And they should allow to differentiate among the target objects and also from the environment [133].

This representation set of features needs to be careful defined (*Feature Engineering* on Figure 1.3) and a training phase, where these features are extracted and learned, is required [133]. The values of the features of each object need to form clusters in the feature space under any possible transformation that the objects may be subject to recognition time. This means the object representation (features that describe an object) needs to be invariant to some real-life transformations.

Some common desired invariances are related to [217]:

- Illumination: non homogeneous intensity changes, depend of light direction, strength and color
- Scale: distance of the object to the camera
- Rotation: rotation around the normal of the projection plane of the camera
- Background clutter: external elements in the background
- Occlusions: interposition of other objects
- Viewpoint: Relative position of the camera in the 3D space.
- Material properties: such as glossy or transparent, that variate depending of the environment and viewpoint
- Surface appearance variation: due to grime or oxide
- Geometrical variations: deformation or intraclass variations

Therefore, this entails the development of a high amount of algorithms over the time, each one with different constraints and requirements [217].

On the other hand *Feature Learning* techniques (Figure 1.3) allow the machine to be fed with raw data and automatically define the features required for discrimination that are invariant to the transformation present in the training set. Further, they have remarkably improved the state of art of OR [136].

This type of techniques, such as Convolutional Neural Network (CNN), is composed by a sequential set of layers. Each one transforms the image representation with non-linearities, starting with the raw input and in deeper layers, a higher level of abstraction of the image representation can be found. In this way complex functions can be learned [136].

Thus, the major efforts in applying this kind of technique are not only in the architecture (configuration of the layers) but also in obtaining the training data. Where a large number of representative labeled data is required for feeding this type of OR technique.

In order to overcome this limitation, different techniques have been proposed to reduce the effort of getting large amounts of manually labeled data: Domain Adaptation (DA) where training phase is supported with labeled data from a related domain for learning a classifier from an unlabeled one [45]. Data augmentation that increases the amount training data with artificially generated samples [59]. And the use of synthetic datasets based on artificial mass produced labeled samples [214].

## 1.5 Motivation

This research has been motivated by two main issues:

- In most of the cases of OR for AR, the implementation relies on the use of *fiducial markers* or a very specific setup or highly constrained situation.

  Marker-based implementations are already well-studied [68], and they suit some applications. But in most cases, it is not possible to use this technique [133]. Additionally, the use of markers will only tell a small information (object position and rotation) about what is happening in the real context.

  Likewise, most *markerless* recognition systems impose constrains regarding the type of elements that are able to recognize. Some, of the more commonly found, use interest points (SIFT [147], ORB [189]) which are available only in textured objects that are not commonly found in industrial elements [90]. Some others are constrained to objects with some characteristics, like the type of surfaces [12, 117], convexity [87] or the presence of some geometry invariants [167].

  Other OR techniques, such as *feature learning*, are able to learn the features of the objects and have shown to work almost with any type of object in any condition [94, 212]. But, one main concern is that they require a lot of labeled training data which in many situations is infeasible to get [45].

  Further, current methods for creating synthetic datasets [114,160,177,183,214,234] do not consider realistic shading or control the variations that are present in real life. And both data augmentation and Domain Adaptation (DA) require to have a labeled or unlabeled dataset of the target objects.

- It is known that there is not an ideal system or configuration, each system depends on the different domain conditions [175].

  Since the first comprehensive review of AR proposed by Azuma 1997 [21], much other research has followed the same steps, gathering different implementations and showing the technological and human-interaction issues. Nevertheless, they show specific implementations, that is, they present different *cases of use* about some specific implementation that worked under some conditions.

However, no relationship among the implementations and neither generalizing the ideas in a global frame is presented, where the knowledge is not linked to other researches. As result, this makes difficult to reuse the knowledge and understand the influence of the different factors in the final implementation.

Thus, nonexistence of a general framework made difficult to classify this information and to re-used it in future implementations.

## 1.6 Thesis Overview

### 1.6.1 Organization of the thesis

The structure of this thesis is as follows: A systematic review of the state of the art and a general framework for IAR is developed in Chapter 2, where four main factors that influence technical implementation are analyzed. Chapter 4 present a method for the generation on synthetic datasets based on the variations present in the domain with a realistic shading approach. In Chapter 5 is presented a series of experiments using the proposed method for the creation of synthetic datasets for training and evaluating *surrounding understanding* methods. Finally, conclusions, limitations and future work are presented in Chapter 6.

### 1.6.2 Contributions

The contribution of this work can be grouped in three main sources:

1. General framework of IAR applications based on the identification of elements of the industry (Domain Variables (DV)) that could affect a technical implementation. In total, 4 Domain factors with 66 variables that influenced 5 implementation factors were identified (Chapter 2).

   This study has been oriented to reach a general understanding of all the variables that could affect an AR implementation and to present some solutions already developed. Also, to propose to developers and researchers a global framework that could help to analyze future implementations by taking into account each one of the variables.

2. DV effect on surrounding understanding algorithms, in this chapter is presented the influence of the DV on technical implementations related to the processes intended to understand the surroundings.This analysis was made by first clustering the process that each one of the DV influences, and also defining what issues cause each one of them. Finally, similar issues caused by the DV (Chapter 3).

3. A method for recreating relevant Domain Variables (DV) using a Physically Based Shading (PBS) approach is proposed, in order to create datasets for training and testing surrounding understanding algorithms. This method is framed under the industrial field, where the parts are very similar, present glossy effects and are subject to processes that change their visual appearance. The method allows generating fully labeled synthetic datasets specifying the distribution of the relevant variables that affect surrounding understanding algorithms (Chapter 4).

4. Ablation study related to the use of the proposed method for the creation of synthetic datasets. The performed study included two types of experiments, one with unknown statistics about the target domain and other with known statistics meaning that some of the characteristics of their parts were known. Further, were found that fine tuning models with proposed method reach comparable (No statistical difference) with models trained with photos. This results validate the proposed method as a viable alternative for training surrounding understanding algorithms applied to industrial cases (Chapter 5).

CHAPTER 2

# Industrial Augmented Reality General Framework

## 2.1 Introduction

For any problem-solving activity it is necessary to define which are the constraints to be satisfied, for instance for either Technical System [173, 185, 221] and software development [204, 215] have been proposed methodologies that include in their process the definition of such constraints, where they are called requirements or, design specifications.

Taking the definition proposed by Ian Sommerville and Pete Sawyer [109] the requirements are "Descriptions of how the system should behave, or of a system property or attribute. They may be a constraint on the development process of the system".

Hence, in similar problems most of the variables that need to be analyzed to identify the requirements are similar, as an example, Stuart Pugh suggested a set of elements ("primary triggers") which are the constituent elements of the Product Design Specification applicable to all products irrespective of the technology [185].

Similarly, there is a set of variables that need to be considered in an IAR problem where there are some variables that remain constant that need to be defined. Therefore, in this section, we propose a set of variables that are required to analyze during the development of an AR implementation.

These Domain Variables (DV) correspond to the characteristics of the industrial task and are not related to the technology but affect the development of an AR application. Further, they map the characteristics and status of the process into fixed parameters that are required to define the technological implementation in a specific situation.

Nevertheless, a fewer research effort until the date has been found in terms of methods that allow a systematical evaluation of the requirements for an implementation of AR in industrial applications. Leading to isolated developments that just show the benefits of the use of AR in specific cases.

Currently, state of art studies related with IAR shown the different cases of application of AR in specific tasks of the industry but fail to show the relation among them and to generalize the different solutions, which make difficult to reuse the knowledge in future applications.

Thus, inspired by some systematical methodologies used both in Technical Systems (TS) and in Software engineer standing that for any solving problem activity is required to define the characteristics that constrain the problem and needs to be satisfied. Therefore, here it is presented an analysis of the state of art focused on to identify which characteristics were constraining their current implementation, what they influenced and how they were solved.

This research is aimed to present a detailed review and a general frame of IAR through the identification of some relevant characteristics of the industrial process (DV) in the previous AR applications. These DV are regarding the user, parts, environment, and task that have an impact on the technical implementation and user performance and perception.

The aim of this research is to give developers or any researcher with interest in the field key elements for implementing AR systems. Also to provide a list of elements of the domain that are required to take into account and how they may interact with this technology.

In the next section, a review of the relevant surveys and another attempts to cluster AR applications are presented (Section 2.2). Later, the method used to analyze the state of art AR applications. In the next Section 2.4 the variables of the domain that have some influence in the implementations. In Section 2.5 the process and properties influenced by the DV. And finally, the conclusions and future work is presented in Section 2.6.

## 2.2 State of the Art

In 1997 Ronald T. Azuma presented one of the most relevant AR surveys that not only will establish one of the most accepted definitions of the technology but showed to community potential applications and issues of AR in six different areas: medical, manufacturing and repair, annotation and visualization, robot path planning, entertainment and military aircraft. And, since then, many other authors have followed the same path of reviewing industrial applications of AR [21].

X. Wang et al. in 2016 reviewed AR-based assembly systems focused on the characteristics and an overview of the technical features of publications between 1990 and 2015 in where a typical AR architecture is composed by six modules: *a*) video capture *b*) image analysis *c*) processing *d*) tracking process *e*) interaction handling *f*) assembly information management *g*) rendering kernel [232].

Further, the AR assembly systems could be classified in three categories: design and planning, operation guidance and training. And the current issues are related to tracking and registration for industrial scenarios (poorly textured objects, bad lighting, smooth surfaces among others) and collaborative interfaces [232].

In 2014 Fabrizio Lamberti et al. presented the challenges and opportunities of AR for maintenance in which two main issues were identified: recognition, tracking, and registration when it is not possible to use markers and dynamic system reconfigurability for instance when unexpected situations occur [133].

Similarly, A.Y.C. Nee et al. in 2012 reviewed AR applications in manufacturing and design based on hardware and software used; activities in the industry of application such as robot path planning, CNN simulation among others; challenges of the technology and human factor and interaction. They conclude that compared with other areas of application, IAR is still relative new due to the higher level of some requirements as the accuracy in registration and tracking and other issues such as ergonomic and human factors [164].

Mauricio Hincapié et al. in 2011 presented examples of AR in maintenance showing their main advantages and flaws. The main advantages are the flexibility of application of AR being suitable for different tasks where it can reduce errors and lower operational costs. However the main disadvantages presented regarding hardware and software such as low-resolution displays, weight, and high computational costs jeopardize their implementation in industry [101].

Additionally, S. K. Ong et al. presented a comprehensive survey of AR development in manufacturing in 2008 intended to show insights for developers and researchers. In which major research in manufacturing, assembly, training, and maintenance was reviewed showing that for an AR application be successful in manufacturing is required to have the next characteristics: *a*) convenient to the user (accurate, smaller, lighter, cheaper, among others) *b*) efficient UIs (User Interfaces) *c*) fast and stable collaborative systems [170].

Moreover, George Papagiannakis et al. categorized the different mobile and wireless technology and their impact in AR to facilitate the understanding of the state of art. They focused on review enabling technologies (hardware, software, registration, and tracking) focused on mobile technologies. They concluded that there is not a unique ideal system approach rather than specific characteristics driven by the domain [175].

In order to compare the proposed AR applications in industry and have a clear taxonomy, Fite Georgel Pierre in 2011 organized the different applications in their life-cycle. Additionally, they proposed a rubric to evaluate the applications that consider: *a*) workflow integration *b*) scalability *c*) cost-benefit *d*) out of the lab status *e*) user tested *f*) out of developers hands *g*) involvement with industry [74].

Their results showed that only a small majority of the projects involve some industrial partner and that of all the studied applications only two have broken out of the laboratory. Additionally that as an emerging technology it is required to be cost beneficial, scalable and involves the companies in their development [74].

In 2015 Huma Shoaib et al. presented a survey of the complete AR area arguing that since 1997 when the first survey was presented [21] there was just incomplete attempts. Additionally, they presented an application-centric review, where the tools used were: Vuforia SDK, MetaIO, ARToolkit, OpenCV, and Kinect. The areas surveyed: healthcare, industry and manufacturing, navigation, mobile AR, education, museums and entertainment [200].

In general, the main goal of AR proposed by Azuma [21] that the virtual objects merge with the real ones somehow is achieved. However, the main limiting issues are still regarding technological limitations, user interfaces, and social acceptance [200].

Similarly, D.W.F. van Krevelen and R. Poelman survey the state of art in both technology and human factors. The current limitations of AR can be grouped into portability, tracking, depth perception overload and over-reliance and social acceptance. And, independently of the technical challenges for AR become part of a user's everyday life, it has to solve issues regarding intuitive interfaces, costs, weight, power usage, ergonomics and appearance [223].

Valerio Elia et al. proposed in 2016 a model based on Analytic Hierarchy Process (AHP) for support a quantitative assessment of AR devices based on features of the manufacturing process and technical knowledge. At the evaluation time, it allows to move from process criteria such as reliability and agility to technical criteria of the characteristics of the AR devices [66].

In 2016 Azuma presented a short paper regarding the most important challenges of AR for being suitable as a natural interface in new technology trends such as IoT (Internet of Things), where the key limitation is the semantic understanding of the surrounding world where two main approaches can be used: object and scene recognition and model the real world beforehand [20].

Summarizing, since the first comprehensive review of AR proposed by Azuma 1997 [21] much other research has followed the same steps, gathering different implementations and showing the technological and human-interaction issues. Nevertheless, they show specific implementations but no relation among them and neither generalizing the ideas in a global frame. As result, this makes difficult to reuse the knowledge and understanding the influence of the different factors in the final implementation.

Additionally, there is not an ideal system or configuration. Each system depends on the different domain conditions [175]. As result, here is proposed a review of AR applications in the industry with the aim of identifying the different domain factors and how these factors have influenced the implementations.

## 2.3 Method

An initial set of domain characteristics based on the authors knowledge were used as starting point. Next step was gathering articles of IAR in scientific databases and previous augmented reality reviews that gather most of the IAR world development.

Mainly, works from 2006 where considered and 70 reviewed papers present an industrial implementation of AR, of which 4 of them were previous surveys. The distribution of articles per year can be seen in Figure 2.1.

Following a study of each article, characteristics of the industrial task were collected, considering if they affect or were taken into account in the development of the AR system. The characteristics that were collected included: controlled conditions and characteristics that the AR system can handle.

Then,the issues caused by each one of the DV were identified. Finally, a clustering process was made in order to find common elements based on a pairing between each domain variable and the influenced factor.

## 2.4 Domain Variables

The general map of the DV and the influenced factors can be seen in Figure 2.2. The main DV belong to four main groups regarding the characteristics of the user, the parts, the environment, and the task. And they influence the techniques or technological process and

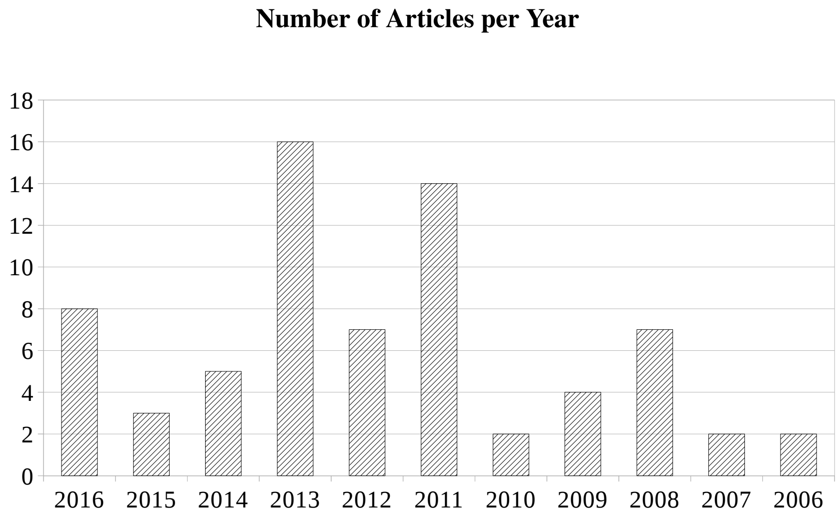


**Figure 2.1:** *Number of articles reviewed per year. Total of 70 articles of AR.*

properties used for understanding the environment and the user, the system characteristics, user related issues and interfaces.

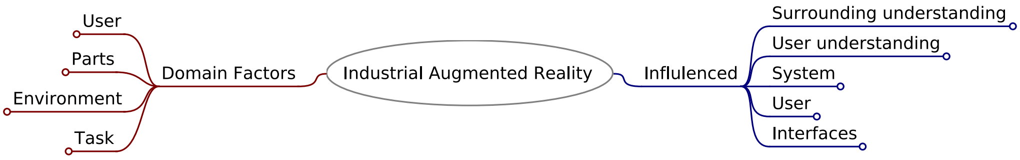


**Figure 2.2:** *General map of DV and influenced factors.*

In this section the DV related to the four categories (user, environment, parts, and task) are discussed. Additionally, all the variables can be found in tables (2.1 - 2.4) where the variables and the possible values or characteristics mentioned by the authors are summarized.

### 2.4.1 User Variables

The main variables related with the user found in literature are shown in Table 2.1. In the next paragraphs will be described the DV and their characteristics.

The variables related to the displacement describe the user movement that is generally performed to accomplish the task. The systems should maximize the range of motion, facilitate the natural and free movements and to be robust to fast and complex movements [26,49].

Similarly, the hands are defined by what is the preferred and more dexterous hand for performing tasks (dominant) and the aim is to ease the use of it [26]. Also, another

**Table 2.1:** *User related DV and values found in literature, both solved or mentioned in the applications*

| | Authors | Variable | Values |
|---|---|---|---|
| **User displacement** | [24, 26, 42, 69–73, 83, 99, 103, 105, 125, 129, 143, 180, 228] | Movement | Fixed \| Mobile |
| | [12, 42, 99, 228] | Velocity | Rapid \| Slow |
| **Hands** | [26, 96, 98, 250] | Dominance | Left \| Right |
| | [129, 139, 250] | Movement | Erratic<br>Fast<br>Motion \| Static |
| | [40, 42, 49, 61, 69–71, 73, 83, 96–100, 128, 139, 161, 167, 167, 169, 226, 227, 236, 237, 250] | General | Shocks<br>Self occlusion |
| | | Dimensions | Range[1] |
| **Skin** | [96, 167] | Color | Variations on pigmentation[1] |
| **Voice** | [97, 167] | Language | Languages[1] |
| **Hearing** | [71, 167, 227] | | |
| **Vision** | [40, 61, 71, 96–98, 218, 227] | Peripheral vision | Range[1] |
| | | Eye dominance | Left \| Right |
| | | Medical conditions | Medical conditions[1] |
| **Cognitive capacity & Skills** | [40, 71, 102, 105, 186] | Spatial ability | High \| Low |
| | | Skills | (Un) Skilled |
| | | Motor skills | Range[1] |
| **Gender & Age** | [102, 103, 194] | Gender | Male \| Female |
| | | Age | Range[1] |
| **Touch sense** | [70, 96, 108, 161, 167, 238] | | |
| **User Experience** | [61, 70, 100, 103, 105, 139, 162, 195, 218, 241] | With task | No experience<br>Little experience<br>Novice |
| | | With AR | No experience<br>Little experience<br>Novice |
| **Ergonomic** | [26, 70, 99, 161, 167, 226, 227] | User Movements | Ergonomic assessments[1] |
| | | Individual measurement | Variations among users[1] |
| **Psychological** | [99, 128] | Social factors | Social factors |
| | | Resistance to change | Range[1] |
| **Familiarization** | [126, 218] | Time using AR | Range[1] |
| | | Resemblance with past tools | Range[1] |
| **Safety & danger awareness** | [24, 61, 96, 99, 227] | Awareness of obstacles | Range[1] |
| | | Standards | Standards[1] |
| | | Guidelines | Guidelines[1] |

variable is about how are the movements of the hands while performing the task or using the hand-held device. Usually, due to the type of tasks performed in industry, allowing the hands to be free and bare hands interaction is preferred or required in some cases [83, 97, 169, 237, 250].

Additionally, the use of haptics in AR is aimed to stimulate the user's touch sense, allowing to touch both real and virtual elements as well as augment real objects with tactile information [115]. In the current research, no variables regarding the user touch sense were mentioned. However other studies in haptic perception show different variables such as part of the body, skin temperature, area of contact, time of exposition, among others [138].

The skin color variations among the users are required to be considered together with lighting changes even in controlled environments. Additionally, other skin color alterations, such as tattoos, could influence the performance of some systems functions.

User voice and hearing were commonly used as a complement communication channel with the system that allows hands free interaction [97]. However, not mayor research or variables were highlighted and, similarly to the skin variations, the environment noise is related to the performance of these two channels.

Furthermore, most of the reviewed articles were primary for augmenting the visual human system. The variables regarding the visual system influence in the general perception and more important of hazard, obstacles, and dangers. Elements such as peripheral vision, that is defined by the angle between the line sight and the location of the stimulus in the visual field, play an important role in the overall user performance [8].

Additionally, vision problems could be reflected in the use of contact lenses or glasses by the user, or require other type of image correction considerations [98]. Other issues such as color blindness could have an impact on how the world is perceived [61].

Moreover, the eye dominance is the preference for the visual input from one eye over the other and it has an important role in stereo vision as the primary source of precise information [121].

The user cognitive capacity is also related to the ability of a person to mentally move into some environment and the visual-spatial imagery manipulation [102]. Further, the spatial ability is among all of the components of intelligence that has been studied most frequently in connection with software use [206].

The cognitive ability is related to the capacity of a user to interpret interfaces and its functions [190], and, together with the user's skills, influences their performance [71].

Regarding the user gender and age, a small research was found in IAR. And their results show that no difference in performance benefits of using AR was found [103, 194].

On the other hand, other studies related to the user age and technology adoption in the workplace of a new software system showed that age plays an important factor in technology adoption, where younger workers presented a more salient attitude towards using the new technology [159].

Similarly, social and cultural factors have an influence on technology acceptance. For instance, the resistance to change deals with the individual inclination to resist changes and could be derived from individual personalities, such as cognitive rigidity, reluctance to lose control, lack of psychological resilience, preference for low levels of novelty, among others [171].

Likewise, the familiarization with the technology is affected by the resemble with previous tools and time exposed to the technology according to the authors. Further, it is likely that the more familiar the user is with an object the more he or she like it. Simple exposure time could affect the people to have more affinity [60].

Besides, previous experience with the task, tools or with AR supposes different levels among the users. Moreover, experience in a job is a crucial factor that influences performance among different age groups. In general, the more work experience the better the performance independently of the age [78].

The mentioned ergonomic issues influence how the system fits the human body. These factors influence directly in the comfort of the users and in their long-term experience.

Finally, none of the elements that compose the AR applications should restrict the user awareness of danger or instruct the user into the performance of some risk action [96,99].

Figure 2.3 presents the number of found references related with each user factor.

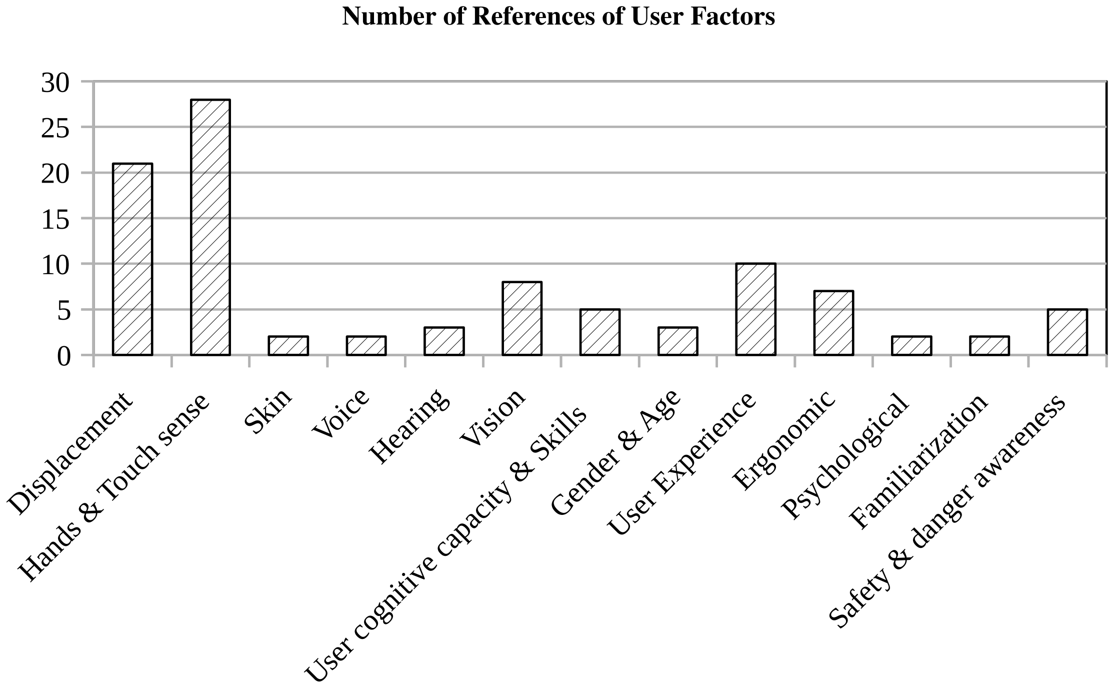


**Figure 2.3:** *Number of references related to the user factors founded in literature. Displacement (21), Hands & Touch sense (28), Skin (2), Voice (2), Hearing (3), Vision (8), User cognitive capacity & Skills (5), Gender & Age (3), User Experience (10), Ergonomic (7), Psychological (2), Familiarization (2), Safety & danger awareness (5).*

### 2.4.2 Parts Variables

Most of the variables in this section are regarding properties that affect the visual appearance and the tactile perception of the parts required for the industrial task. The complete list variables related to the parts can be found in Table 2.2.

Despite that the appearance of the objects depends of complex interactions between light, geometry and material properties [184]; the large number of optical properties in

**Table 2.2:** *Parts related DV and values found in literature, both solved or mentioned in the applications*

| | Authors | Variable | Values |
|---|---|---|---|
| **Material** | [90, 97, 98, 100, 143, 195, 250] | Reflection | Specular<br>Diffuse |
| | [12, 12, 49, 90, 117, 117, 129, 143, 195, 228] | Texture | Minimal<br>Low contrast<br>Blurred<br>Distribution ranges [1]<br>Frequency ranges [1]<br>Unique<br>Textureless |
| **Shape** | [12, 42, 87, 90, 103, 105, 117, 143, 167, 169, 228, 240] | Surfaces | Planar<br>Cylindrical<br>Conical |
| | | Convexity | (Non) Convex |
| | | Invariants | Concavities<br>Collinear<br>Tangency<br>Parallelism |
| | | Boundaries | (Non) Homogeneous surface |
| | | General | Self-occluded<br>Complex geometry<br>Number of features [1] |
| **General Appearance** | [162] | Relevance (to user) | Relevant \| Irrelevant |
| | [12, 40, 42, 49, 61, 69, 71, 86, 87, 90, 98–100, 103, 117, 125, 139, 143, 167, 180, 186, 228, 240, 250] | Occlusion | Frequent<br>Permanent<br>Partial |
| | [24, 49, 86, 86, 87, 87, 117, 143, 180, 186] | Affine transformations | Scale<br>Rotation<br>Translation |
| | [12, 24, 49, 49, 87, 87, 100, 143, 167] | General dimensions | Large \| Small<br>(Non) Planar |
| **Set** | [167] | Equal parts | Unique<br>Similar<br>Equal |
| | [86–88, 99, 100, 117] | Number of parts | Range [1] |
| | [126, 167] | Arrangement | Random<br>Overlap<br>Stack |
| | [103, 105] | Dimensions | Disparity \| Homogeneous |
| | [24, 61, 90, 90, 105, 162, 167] | Color | Disparity \| Homogeneous |
| **Appearance change** | [100] | Movable part | Range [1] |
| | [143, 180, 250] | Deformations | Small \| Large<br>Rigid |
| | [228, 250] | Incomplete | Partial<br>Out of view<br>Missing parts |

solid state materials can be classified into small number of general phenomena that are: *reflection*, *propagation* and *transmission* [75].

Further, the light reflection can be decomposed into two: *specular* reflection, that occurs when the incident parallel light rays meet a smooth surface and (the light rays) are reflected similarly parallel. And, on the contrary, when light meets an uneven surface light, rays are reflected in many directions and it is known as *diffuse* reflection [25].

Another variable related to the material that affects the object's appearance is the texture. The object texture refers to the intensity variations along the object geometry without taking into account the effects of lighting. In which the different values of this variable indicate the characteristics of the texture, for instance, blurred or with low contrast.

The characteristics of the shape (geometry) refer to a specific configuration of the external boundary of the objects that have special properties. Most of these properties are regarding the invariance of the projection to a plane from different Point Of View (POV) (projective transformations) such as collinearity, tangency or parallelism [87].

Other shape characteristics allow defining assembly relation among parts such as planar, conical or cylindrical surfaces [169]. Further, the number of these characteristics is relevant to make some inferences.

As a whole, the general appearance perception could be affected by the affine transformation between the observer (sensor) and the part, different levels of occlusion and parts general dimensions that could be large or small enough to fit into the sensing or viewing range. Also, this general appearance perception could be important for the user.

Additionally, when one of the general dimension of the part is significantly smaller than the other two, it can be represented as a 2D or planar object [87, 143, 167].

Furthers there are some properties of the parts that could change the general appearance from a POV. For instance, parts with movable components that can freely translate or rotate, deformable materials or structures such as springs or foams and incomplete parts due to external factors.

On the other hand, the relation among the parts that compose the task is grouped into the set factors. The similitude in different characteristics such as color or dimensions plays an important role. Also, the arrangement about how the parts are located regarding each other and number of parts that compose a specific task.

Figure 2.4 presents the number of found references related with each parts variables.

### 2.4.3 Environment Variables

The environment variables are issues regarding the element and conditions of the surrounding. All the variables are listed in the Table 2.3.

The working area variables are related to the ambient conditions. Temperature and noise are expected to have different ranges that could be controlled and/or present a homogeneous behavior.

Similarly, the environmental conditions describe what it is expected in a typical manufacturing environment such as electrical noise, dirty, dust, among others [69, 96]. Likewise, different lighting conditions are expected, from monotonous homogeneous to outside (natural) lighting conditions. This type of characteristics made up a hazardous environment which imposes critical device requirements [96].

The background is usually related to other elements that are not of main interest to the task, and their characteristics are fundamental for separating it from the interest objects.

**Table 2.3:** *Environment related DV and values found in literature, both solved or mentioned in the applications*

| | Authors | Variable | Values |
|---|---|---|---|
| **Working Area** | [61, 218] | Temperature | Controlled<br>Homogeneous<br>Temperature ranges [1] |
| | [70, 218] | Noise | Controlled<br>Homogeneous<br>Noise ranges [1] |
| | [69, 97, 129] | Surrounding variability | Relative static<br>Static<br>Permanent change |
| | [24, 42, 61, 69, 70, 72, 96, 99, 103, 143, 167, 168] | Environment Conditions | Electrical noise<br>Dust<br>Dirty<br>Electromagnetic interference<br>Vibration<br>Grease<br>Weld sparks |
| | [26, 42, 49, 61, 69, 71, 97–99, 129, 141, 161, 167, 197, 228] | Working Area Size | Large \| Small |
| | [12, 49, 90, 96, 98, 143, 167, 192] | Background | Clutter ranges [1]<br>Textured ranges [1]<br>Color Similitude<br>Repetitive structure<br>Clear background |
| **Lighting** | [26, 42, 69, 70, 73, 96, 99, 143, 180, 218, 236] | Variability | Homogeneous<br>Natural variable (Outside)<br>Intensity changes [1] |
| | [69, 71, 87, 99, 100, 195] | Intensity | Range of lighting [1]<br>Monotonous |
| **Knowledge a priori** | [26, 42, 61, 70, 97, 99, 103, 169] | Surrounding | (Un) Expected<br>(Un) Prepared<br>(Un) Controlled |
| | [70, 96, 99, 167, 218] | Lighting | (Un) Expected<br>(Un) Prepared<br>(Un) Controlled |
| **External elements** | [42, 96, 99, 129] | General | Similarity to task parts<br>Resemble buttons<br>Self occluding |
| | | Static surfaces | Linear<br>Curved<br>Planar |
| | | Distinguishable by user | Intersection of hard edges<br>Dimples<br>Small holes<br>Raised geometry |
| | | Movable | Slide<br>Bend<br>Rotate |

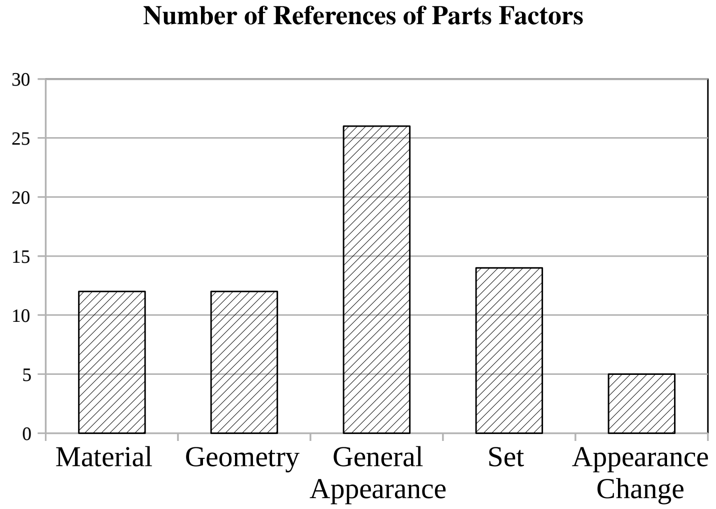


**Figure 2.4:** *Number of references related to the Parts factors found in literature. Material (12), Geometry (12), General Appearance (26), Set (14), Appearance change (5).*

Usually, industrial scenarios can be described as cramped, cluttered with more parts that the ones that are required [90, 98, 180, 192].

Finally, all these variables could present some variation over the time, from a static behavior up to permanently in change.

Further, a priori knowledge of the variables could be obtained in some cases as it is the cases of controlled and expected behaviors. However, in literature, the lighting and the surroundings were the most related factors. And, besides, critical situations arrive in uncontrolled environments [70].

Additionally, shape characteristics of the objects surrounding could be of interest, because they can be used as a support for other functionalities of the system. Therefore, they limit the surrounding geometries that can be presented [99]. In general, they are regarding the shape characteristics such as their surface geometry, and features that could be relevant to the user to make them distinguishable. Similarly, parts that allow user motion could provide user feedback.

Figure 2.5 presents the number of found references related with each environment factor.

### 2.4.4 Task Variables

The task variables are categorized into three groups, the characteristics of the task and parts that affect the complexity, the variables which affect the user performs, and required information to accomplish the task. Besides, all the variables are presented in Table 2.4.

The complexity is an inherent property of the systems, here we refer instead to the *perceived complexity*. The perceived complexity is an observer-dependent property that describes the observer to understand the system [202]. Further, the complexity of the task should be high enough for the user of AR to be perceived as it worth the use of it [211].

Therefore, the suggested variables that could affect the user perception of complexity are related to both parts and task properties, such as the presence of *significant* parts,

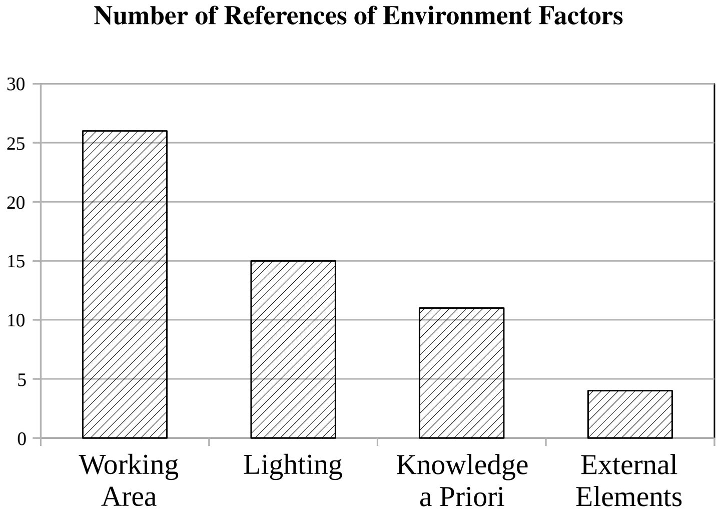


**Figure 2.5:** *Number of references related to the Environment factors founded in literature. Working Area (26), Lighting (15), Knowledge a priori (11), External elements (4).*

which are the ones that produce main functions and, therefore, require more difficult task activities than less significant ones [186].

Other factors that influence the difficulty of the task performance regarding the parts are related to hidden parts and the number of degrees of freedom of alignment of the parts [186]. Similarly, task ambiguity or complexity affects the amount of information required for the user [97].

Additionally, variables related to the user perform are the error awareness, meaning that according to the task it is necessary to consider user errors, where the most common issues related are error detection and prevention [100, 139, 167, 240].

Likewise, the operation time is a requirement that needs to be considered in order to provide uninterrupted support [26]. Furthermore, in the long term use, the strain caused by technology issues that could be intensified by long periods of use should be considered [218].

Finally, the different type of information could be required to fulfill some tasks, therefore, simulation of different characteristics is necessary. For instance, the weight of the parts, the required movement to perform a task, the appearance of the part, among others. Additionally, the characteristics of the used tools should be considered in order to be recognized by the system.

Figure 2.6 presents the number of found references related with each task variable.

## 2.5 Influenced Process and Properties

The previous factors impact several areas of an AR implementation. These characteristics of the domain could affect also other relevant issues such as ethical or legal, which are out of scope. Therefore, this research is focused on how they affect the technical implementation and user perception.

In this section, it is presented how the different domain factors found in literature influence some implementation characteristics according to the authors. The processes and

**Table 2.4:** *Task related DV and values founded in literature, both solved or mentioned in the applications*

| | Authors | Variable | Values |
|---|---|---|---|
| **Complexity** | [96–98, 100, 103, 128, 168, 186, 194, 197] | Parts | Significant<br>Hidden<br>DOF of alignment |
| | | Task | Ambiguity<br>Order |
| **Perform** | [100, 139, 140, 167, 195, 238, 240] | Error Awareness | Wrong position<br>Prevent continue in error<br>Error feedback<br>Correct action |
| | [26, 218] | Time | Time ranges [1]<br>Uninterrumped<br>Long term use |
| **Information** | [12, 71, 102, 105, 125, 192, 236] | Simulation | Weight<br>Movement<br>Appearance<br>Machining<br>Assembly<br>Paths<br>Interferences<br>Disassembly |
| | [12, 71, 83, 97–99, 169] | Tools | Arrangement<br>Digitalized<br>Similar shape to task objects<br>Considered object of interest |

properties of the implementation that are influenced by the characteristics of the domain can be grouped into five general categories: *a*) Surrounding understanding *b*) User understanding *c*) System *d*) User perceptions and performance *e*) Interfaces.

A summary of the implementation categories and the variables that have an influence upon them can be seen in Figure 2.7.

Since the main field of interest of this thesis is the surrounding understanding, it is presented in their own Chapter 3

### 2.5.1 User understanding

The User Understanding is a set of processes aimed to recognize, segment and interpret the user of an AR system. Five main processes were identified: user recognition, tracking, sensing, segmentation and task awareness.

As well as the Surrounding Understanding (Chapter 3) the boundaries that divide this process in real implementations are diffuse, hence, they are treated as general processes and not specific implementations.

The different processes influenced by the DV of the User Understanding can be seen in Table 2.5. As expected, the process related to the user understanding is less influenced by the characteristics related to the parts and more influenced by the user and environment characteristics.

Similarly to the Surrounding Understanding, the DV cause *issues* in the process but also constrain or impose *boundaries* to the used techniques. The main issues generated by the DV are shown in Figure 2.8 where the features are related to the characteristics of

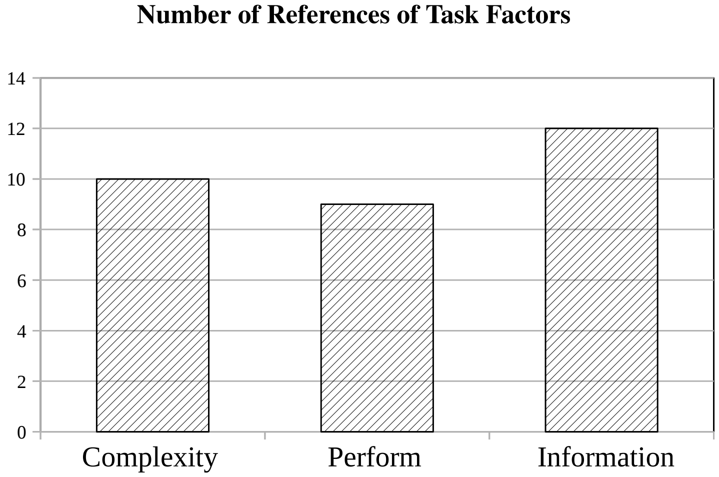


**Figure 2.6:** *Number of references related to the Task factors found in literature. Complexity (10), Perform (9), Information (12).*

**Table 2.5:** *Domain Variables that affect each one of the subprocess of the user understanding according to authors.*

| **Domain Variables** | | **User Understanding Process** | | | | |
|---|---|---|---|---|---|---|
| | | Recognition | Tracking | Sensing | Segmentation | Task Awareness |
| User | Movement | ■ | | ■ | | ■ |
| | Hands | ■ | ■ | | ■ | |
| Environment | Ext. elements | | | | ■ | |
| | Area size | | ■ | ■ | | |
| | Lighting | | | | ■ | |
| Task | Errors | | | | | ■ |

the user, such as skin or hand shape.

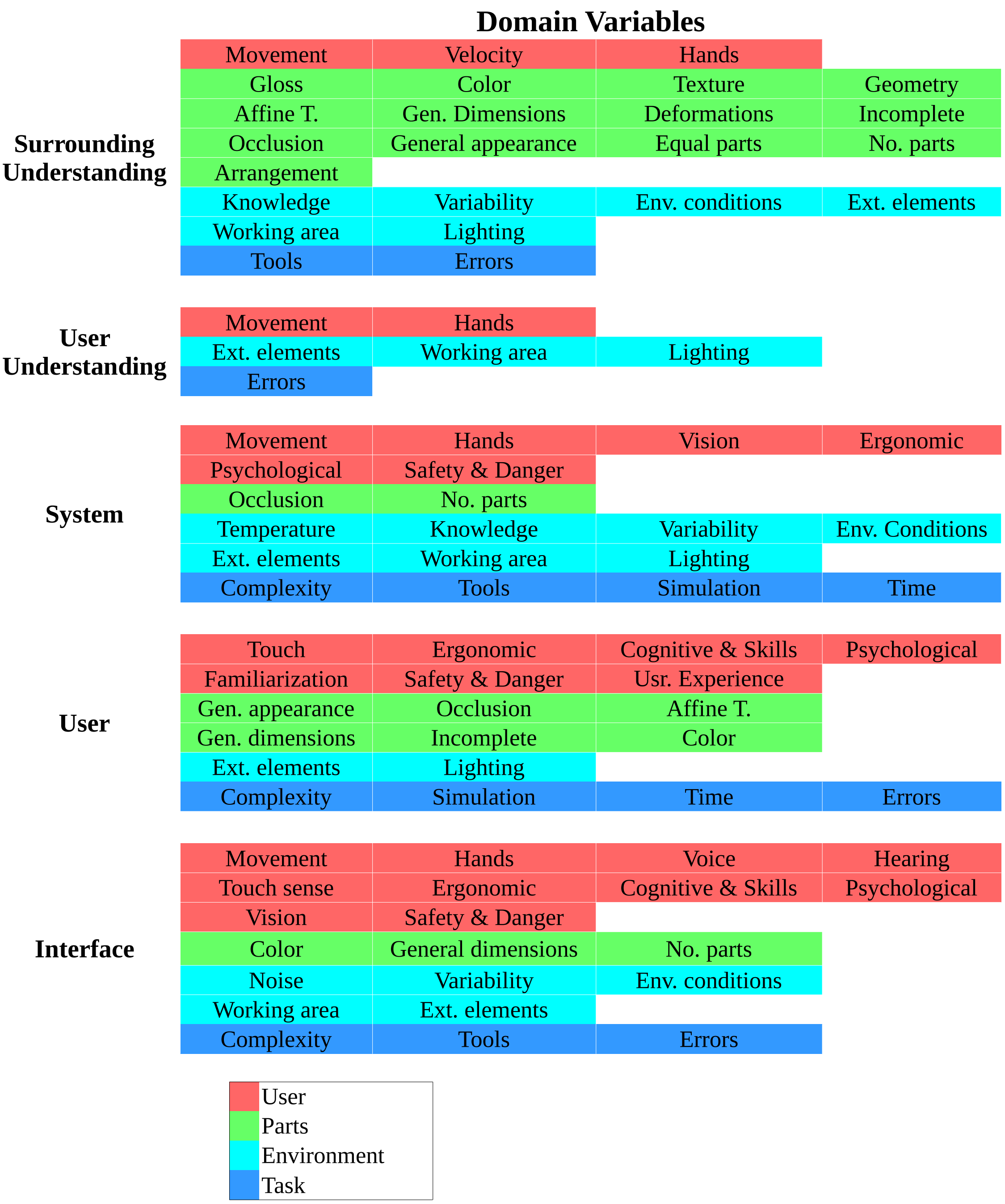
Therefore, the main *issues* caused by the DV can be grouped into:

**Confusing actions.** When interpreting the user, some approaches are based on the patterns of the user motion. For instance, hands gestures that interact with virtual information. Some patterns of motion do not necessarily indicate a wanted interaction. Fiorentino proposes a virtual area in space where the hand gestures of the user are recognized as intentional interactions [70].

Another approach for understanding the user is based on the errors that make while performing an activity. Thus AR instructions are presented accordingly to the user abilities. However, a withdraw of this approach is to rely on the object recognition that is influenced by the previously presented DV [240].

**Distortion of the features.** Similar to the Surrounding Understanding, the characteristics used to understand the user can be distorted by external elements. Fast movements of the user and variations of lighting could affect the tracking of the user.

**Domain Variables vs. Influenced Process and Properties**



| | Domain Variables | | | |
|---|---|---|---|---|
| **Surrounding Understanding** | Movement | Velocity | Hands | |
| | Gloss | Color | Texture | Geometry |
| | Affine T. | Gen. Dimensions | Deformations | Incomplete |
| | Occlusion | General appearance | Equal parts | No. parts |
| | Arrangement | | | |
| | Knowledge | Variability | Env. conditions | Ext. elements |
| | Working area | Lighting | | |
| | Tools | Errors | | |
| **User Understanding** | Movement | Hands | | |
| | Ext. elements | Working area | Lighting | |
| | Errors | | | |
| **System** | Movement | Hands | Vision | Ergonomic |
| | Psychological | Safety & Danger | | |
| | Occlusion | No. parts | | |
| | Temperature | Knowledge | Variability | Env. Conditions |
| | Ext. elements | Working area | Lighting | |
| | Complexity | Tools | Simulation | Time |
| **User** | Touch | Ergonomic | Cognitive & Skills | Psychological |
| | Familiarization | Safety & Danger | Usr. Experience | |
| | Gen. appearance | Occlusion | Affine T. | |
| | Gen. dimensions | Incomplete | Color | |
| | Ext. elements | Lighting | | |
| | Complexity | Simulation | Time | Errors |
| **Interface** | Movement | Hands | Voice | Hearing |
| | Touch sense | Ergonomic | Cognitive & Skills | Psychological |
| | Vision | Safety & Danger | | |
| | Color | General dimensions | No. parts | |
| | Noise | Variability | Env. conditions | |
| | Working area | Ext. elements | | |
| | Complexity | Tools | Errors | |

User
Parts
Environment
Task

**Figure 2.7:** *General view of Domain Variables (user-red, parts-green, environment-clear blue, task-dark blue) and their influenced implementation process.*

**User Understanding Main Issues**

| Issues | Domain Variables | | |
|---|---|---|---|
| Confusing actions | Movements | Errors | |
| Distortion of the features | Movements | Ext. elements | Lighting |
| Fake features | Ext. elements | | |
| Incomplete features | Hands | | |

🟥 User
🟩 Parts
▧ Environment
🟦 Task

**Figure 2.8:** *Main issues generated by the DV (user-red, parts-green, environment-clear blue, task-dark blue)that influence User Understanding implementation in AR.*

Henderson and Feiner [99] propose the use of hybrid inertial-optical tracker that fusion inertial data with optical markers information, making it robust to movements and intermittent lighting.

Likewise, color changes in lighting affect segmentation techniques that are based on the color of the skin. The controlled environment has been proposed [96, 99, 127].

Further, tracking systems that rely on magnetometers can interact with ferromagnetic elements making it not suitable for most industrial tasks [99].

**Fake features.** Elements of the environment or the parts of the industrial operation that have similar characteristics to the ones used in the processes for understanding the user.

Objects that have the same color as the skin affect segmentation [96], shape for gesture classification [249], or other properties (magnetic) [99].

**Incomplete features.** As mentioned before, there are approaches for understanding the user by seeing what is (s)he doing with the objects.

Petersen and Stricker [180] proposed a framework for automatic define the step-by-step documentation of a manual task from a video. Thus, it is required to handle permanent occlusions that can be made by the user during the performance of the task hiding useful features in task recognition. Instead of using OR, they propose to use a robust distance function between frames of the video.

Besides the presented issues, constraints are imposed by the DV to the techniques used for User Understanding. The main requisite regarding the use of AR in industrial applications is about not constraining the free movement of the user with external hardware as much as possible.

This involves, avoid the user to look for specific directions while performing the task. Add external cameras to keep track of the user body of other elements if the input of the system is from the user point of view [96]. Similarly, other sensors can be used for this task [83].

Also, mechanical based tracking systems to determine the position of the user body parts constrain the user free motion being unsuitable for industrial tasks [99].

The second constraint is regarding the preference of the use of one side of the body over the other (Laterality). For instance, the installation of devices needs to take into account the hand dominance in two ways. The first one is to avoid to place devices that hinder the user in the performing of the task.

The second one is to place devices in the dominant hand to ease the understanding of the user. As an example, Zhang et al. [250] install a wireless RFID reader in the dominant hand of the user to detect the interaction with objects with RFID tags.

Further, many segmentation approaches may rely on user skin pigmentation [96], but other elements can cause unwanted interactions such as tattoos or nail polish that constrain the use of such techniques.

### 2.5.2 System

Almost every AR application is supported by the use of hardware. In this section, is presented how the DV constrain the characteristics of the used devices.

The influenced characteristics of the AR devices most relevant found in the literature that are presented here are regarding size, displays, mobility, layout, autonomy, weight and performance. And how they are related with the DV are shown in Table 2.6

**Table 2.6:** *Domain Variables that affect the main system characteristics according to authors.*

| **Domain Variables** | | **System Characteristics** | | | | | |
|---|---|---|---|---|---|---|---|
| | | Size | Display | Layout | Autonomy | Weight | Performance |
| **User** | Movement | ■ | | | | ■ | |
| | Hands | | ■ | ■ | | | |
| | Vision | | ■ | | | | |
| | Ergonomic | ■ | ■ | ■ | | ■ | |
| | Psychological | ■ | ■ | ■ | | | |
| | Safety Awareness | | ■ | ■ | | | |
| | Occlusion | | ■ | | | | |
| | No. parts | | | | | | ■ |
| **Environment** | Temperature | | | ■ | ■ | | ■ |
| | Knowledge | | | ■ | | | |
| | Surr. Variability | | | | | | ■ |
| | Env. Conditions | | | ■ | | | ■ |
| | Ext. elements | | ■ | | | | |
| | Working area | | ■ | ■ | | | ■ |
| | Lighting | | ■ | | | | |
| | Complexity | | ■ | | | | ■ |
| | Tools | | | ■ | | | |
| | Simulation | | | | | | ■ |
| | Time | ■ | | | ■ | ■ | |

The requirements for IAR devices based on the domain characteristics are as follows:

**Size** The size of the AR devices always has to search for the user comfort. Three types of device size cases were found: small for portability, fit user anatomy and large enough to be comfortably used.

In industrial tasks, long time of use and user mobility are expected. Bulky hardware result in an experience of fatigue [161].

Also taking into account that all of the elements held by the user should be portable and for easy maneuverability such as portable batteries [26].

Further elements attached to the human body needs to fit the human anatomy. Zhang et al. [250] attached a RFID antenna in the user's hands but small enough to not disrupt any assembly operation.

Moreover, the size of projective displays and screens need to take into account the distance of the user during operation in order to allow to comfortably see the virtual elements [71].

**Display** The display is a key element in AR, it delivers the final experience to the user and it has been an area of great interest over time.

Given the context of being used in an industrial field, one of the most suggested requirements regarding the displays is to allow the user to have the hands free.

Therefore, the use of tablets or any holding screen is not that well suited for many industrial tasks [40, 227]. Alternatives include Head Mounted Displays (HMD) [83, 250], augmented desktops, augmented workbench and shared workspace [69], projective displays [99], and magic windows that consists of a touch screen mounted in joint arm [128].

On the other hand, HMD have also known for being bulky and uncomfortable in the long-term use, where still further research is required for the development of lighter AR glasses [128, 167, 227].

Another important characteristic of the displays that are relevant depending on the level of detail required in the task is the display resolution. This characteristic determines the level of detail presented in the virtual content in optical see-through displays and the detail of both real and virtual in video see-through [99].

Further, optical see-through presents the advantage of not down-sampling by directly displaying it the reality in contrast with video see-through [96]. Besides, computer monitors have been shown good results in an assembly prototype regarding display resolution and also, they do not down-sample the reality [105].

Another critical issue, especially in see-through devices, is their brightness. If the brightness is less than the light coming from the environment, the virtual elements are difficult to see. Filters have been used to compensate this issue in the displays, but make difficult to see the real environment. Thus additional lighting needs to be added to the environment [99].

Similarly, the use of projectors as spatial displays (project the information directly into the real objects) may require a controlled lighting and a good contrast with the environment light [167].

Also, with the use of projective displays is necessary to consider the surface where the information is projected, and, both, the shape [99] and color [61] to guarantee good contrast and no deformation. And, also consider possible occlusions of the projection with other elements [40].

And, on contrary to HMD less visual fatigue is presented with the projective displays [40].

Further issues regarding the display technology should consider some medical visual conditions of the user, such as the use of glasses or contact lenses that requires devices calibration [98].

**Layout** Related to the architecture or configuration of the parts of an AR system. The location of the different components of the system are mostly oriented to: map the working environment, allow free user motion and comfort, user laterality, security.

The placement of sensors or cameras is oriented to cover the required areas of work. Fiorentino [71] used cameras according to their functions: fixed pointing at the assembly part and tools and personal camera handled by the user. This entails trying to not hinder the mobility around the area, for instance, ceiling setups have been used [61, 99].

Also, a common arrangement is egocentric system, where the input camera is attached to the users, allowing to see from their point of view [49, 71].

The architecture of the AR is required to ensure the user free motion as mentioned before, wired connections are better to be avoided [71] as well as any mechanical elements attached to the user [99]. And finally, avoid as much as possible any type of devices that user is required to hold or wear [61].

These considerations are made with the final goal of the user comfort. Where the location of the devices must search the use of the dominant part of the body. For instance, Behzadan et al. [26] proposed the location of miniature keyboard positioned on the opposite dominant hand.

Lastly, security considerations about the hardware. All the wired connections need to be thermal and electric isolated in high-risk sites and protected against dust and dirty [96, 103]. Other adversary environment characteristics include electromagnetic interference, vibration, heat, grease, and sparks. Therefore, adequate cooling and isolating systems are required [61].

**Autonomy** Systems are intended to have extended use with uninterrupted operations. Critical design considerations are regarding power source where it is not feasible for the user to change the batteries especially if each component has one different [26].

Additionally, a cooling system to resist high temperatures and prolonged use times [61].

**Weight** Lightweight hardware is one of the issues that has been an obstacle in the spreading of AR especially HMD [99]. Also, any wearable device has to consider the ergonomic analysis and weight distribution in order to avoid user fatigue after a prolonged time of use [26, 161].

**Performance** Main sources of resources consumption are related to large features size, searching space and datasets used in OR, tracking, and registration.

Large features descriptors can burden the memory (Ferns) or be computationally expensive (SIFT). Several approaches to feature descriptors have been developed for the use of mobile devices. Wagner et al. [228] proposed a modified version of SIFT and Ferns that can be used in mobile phones for tracking.

Hagbi et al. [87] proposed a descriptor based on the 2D contour concavities that provides accurate and real-time registration in mobile phones.

Another approach suggests the parallelizing of the process. On one thread track, the erratic motion of the mobile device and on the other thread produce the 3D map features. This provides quality tracking for small textured workspaces that are relatively static [129].

A similar approach has been suggested by Ha et al. [86] but instead the process is performed in a server-client architecture, the mobile device captures the features and the heavy computations are performed on the server side.

Additionally to expensive computational features, the number of objects to be considered increase the complexity of the computations. Jo and Kim [117] propose to only consider the objects that could be present at some locations by using IOT.

Further, the requirements of different levels of accuracy mean more or fewer data processing complexity. And this finally depends on the type of the task and case dependent of how much errors could affect the performance of the activity. Thus, generic interfaces that allow having different modules of accuracy have been proposed regarding GPS positioning [26].

For instance, complex procedures require more detailed information to be presented [186]. Large 3D models require large amounts of memory and also computational power to be rendered in real time. Hakkarainen et al. [88] propose as well a client-server architecture in which the server is in charge of all complex model rendering and the image is delivered to the client.

In both server-client architectures the communication of large 3D assets remains unsolved and it is expected that 3D object streaming be a possible solution [86, 88].

### 2.5.3 User Perceptions and Performance

These are related to how some characteristics of the domain have influenced the user perceptions of the interaction with AR in previous applications. Also, how they influenced the performance of executing a task and the long-term use of AR.

Three main elements of the user perception were identified: the user experience, user performance and the long-term use of the AR. The DV that influence them are shown in Table 2.7.

**User Experience** User perceptions are about what the users think about the system and how they feel after using it. The main elements that influence the user perceptions are related to the user familiarization with the technology, the comfort of the hardware, and the coherent stimulation of the senses.

Familiarization of the user with the task and AR not only influences the performance but also because the user is aware of the limitations of the technology knows how to

**Table 2.7:** *Domain Variables that have affected the user perception and performance of performing a task with the use of AR.*

| **Domain Variables** | | **User Perceptions** | | |
|---|---|---|---|---|
| | | Experience | Performance | Long term use |
| **User** | Touch | ■ | ■ | |
| | Ergonomic | ■ | ■ | |
| | Cognitive & Skills | | ■ | |
| | Psychological | | | ■ |
| | Familiarization | ■ | ■ | |
| | Safety & Danger | | | ■ |
| | User. Experience | ■ | ■ | |
| **Parts** | Gen. appearance | ■ | | |
| | Gen. dimensions | | ■ | |
| | Occlusion | | ■ | |
| | Incomplete | | ■ | |
| | Affine T. | | ■ | |
| | Color | | ■ | |
| **Environment** | Ext. elements | | ■ | |
| | Lighting | ■ | | |
| **Task** | Complexity | | ■ | ■ |
| | Simulation | ■ | ■ | |
| | Time | ■ | | |
| | Errors | | ■ | |

use it. For instance, the awareness of markers occlusion hinder the user interactions [139].

Further, at the beginning of the use of AR higher levels of user strain can be observable compared to the one at the end of the task. These levels of strain when using AR to execute a task are even higher than using another tool such as paper instructions. These findings suggest that AR needs more time of familiarization than classical approaches [218].

Moreover, the coherent use of stimulation of the human senses can help to have a better understanding of instructions, having more immerse experience and strengthen the natural user interaction.

For instance, the use of haptic feedback can help the user to have a more immersive experience when interacting with virtual controls [70, 161].

Further, a possible implementation for having tactile feedback without requiring additional hardware is by using passive haptic feedback, where the virtual elements are loaded over real ones with the same geometry, thus the user can touch these virtual elements. This technique has proved to have beneficial results regarding performance and user acceptance [96].

Also, the use of vibrotactile feedback can give clues to the user about the performance of translational or rotational movements. Webel et al. [238] propose the use of vibrotactile bracelet to communicate the direction of rotation (clockwise or anti-clockwise) of an action that is difficult to see using another type of media.

Other implementations propose the use of visual simulation of the movement required to perform a task. This type of simulation encourages the user to mimic the actions changing the perceived task complexity and has a decreasing effect on the cognitive load [105].

Finally, another variable that influences the user experience is the ergonomic considerations, where devices should accommodate to the user. Avoiding heavy devices hold by the user, low display contrast caused by environment lighting. This minimizes the feelings of tiredness after the use of AR [99, 161, 192].

**User Performance** These are the DV that affect the performance of the user when executing a task using AR. Several elements could have an impact on the user performance such as psychological, cognitive or motor skills. Here will be presented some of them that were expressed by the authors regarding the conditions where a user can have the most benefits of using AR.

The main issues developed in this section related with the performance of the user are the use of haptics, user experience with the task and with AR, the characteristics of the task, user skills and cognitive capacity, and the type of AR.

The use of haptic feedback to support user guidance has been already studied and reported to present benefits [91, 220]. Also, as enhancing the AR experience, for instance, Murakami et al. [161] reported that the average of success performing an assembly task was 5.6 times with haptic feedback compared with 2.5 times without using their proposed system.

Likewise, it is useful to communicate extra information to the visual, such as notify the user if he or she has performed the right action. This is an important factor in the prevention of errors at the initial steps of task [238].

This is also related to the type of information presented in the AR (simulation and errors awareness on the DV). The presentation of the 3D models and animations, of how to place parts in the right position and orientation, makes easier to follow the instructions, being one of the most relevant features of AR [192].

Similarly, error prevention, avoid the user to continue the task with an incorrect solution [167, 195, 238] but also when the user is corrected using an Intelligent Tutoring Systems (ITS), improves the user learning by 25% and was 30% faster compared with AR without intelligent support [241].

Another element is that the benefits of AR are more remarkable when the task is more complex and difficult to the user. And some reported elements that cause the task complexity are linked with degrees of freedom of parts alignment and hidden parts [186], parts with similar length and different shapes (in pipes) [105], the presence of external parts that are not required in task [192], shape and colors disparities [103].

Now, regarding the user, having experience regarding with both, the task and AR influence the performance. Unexperienced users with respect to the task present more benefits of using AR.

In one experiment, exploring the benefits of using AR versus paper instructions in the replacement of a hard disk of a computer, results show that subjects that are used to assembly computers take more time with AR than with paper instructions. But regarding subjects nor related with the task field a reduction of time was evidenced. Further, only one subject has experience with AR [195].

In contrast, experience with the use of AR could present advantages in the performance, due to the user knows the limitations an consideration of the technology. Thus, depending on the case, experience is requested in order to avoid altering the data because the technology is new and strange to users [218].

Therefore, for maximizing the effects of AR, users with experience with the technology, but in complex tasks where they have little experience, could potentially show better improvements regarding classical methods.

Additionally, Hou and Wang [102, 104] propose to discriminate the workload of subjects using AR in an assembly task by their spatial cognitive capacity. The spatial cognitive capacity is related to the ability of motion in mental space (mental rotation) but also considers individual strategies, academic background and practice effect, among others. This allows defining a baseline of the subjects in this domain.

**Long Term Use** Related to the DV that may affect the usage of IAR in the long term. When proposing a new system, especially a disruptive one as AR, there is always a risk of rejection by individuals or the whole community. A system is always prone to have changes due to social factors and the instinct to resist to change. Particularly, in a time constraint with zero tolerance to errors areas. One solution may be to include the user in the design process to increase acceptance chance [99].

For instance, in the system proposed by Kleiber and Alexander [128], HMD was initially suggested for allowing users hands-free. But this type of device had a lack of acceptance by the personnel changing the design for using a touchscreen with a joint arm. Although improvements in HMD have been done over the years still a risk of rejection exists, by safety considerations, for example that this kind of devices limits the user awareness of obstacles and dangers [99].

Additionally, the basis of acceptance of any new technology is the perception of usefulness [133], in which AR has to be cost-beneficial convenient, scalable and reproducible [74].

### 2.5.4 Interfaces

Although AR is considered a type of interface, a Natural User Interface (NUI), different strategies can be used to allow the user to interact with the system. Thus, this section is dedicated to the strategies used in IAR based on the influence of DV.

The DV influence the interfaces by defining the type of interaction used and the type or what information is presented. The summary of this relationship can be seen in Table 2.8.

**Type of interaction** In this section are presented the different strategies used in AR:

- Tangible interfaces: They use elements of the environment as input or interaction devices with the system.

  Considerations are regarding the type of element used because industrial elements can result harmful to the user when being touched such as sharp or hot

**Table 2.8:** *Domain Variables that have affected the user interfaces strategies with the use of IAR.*

| Domain variables | | User Interfaces | |
|---|---|---|---|
| | | Type interaction | Type of information |
| User | Movement | ■ | ■ |
| | Touch sense | ■ | |
| | Vision | | ■ |
| | Hands | | |
| | Ergonomic | ■ | ■ |
| | Safety & Danger | ■ | |
| | Voice | ■ | |
| | Cognitive & Skills | | ■ |
| | Hearing | ■ | |
| | Psychological | ■ | |
| Parts | Color | | ■ |
| | General dimensions | | ■ |
| | No. parts | | ■ |
| Environment | Noise | ■ | |
| | Working area | ■ | |
| | Variability | | ■ |
| | Ext. elements | ■ | ■ |
| | Env. conditions | ■ | |
| Task | Complexity | | ■ |
| | Tools | | ■ |
| | Errors | | ■ |

surfaces. The system that automatically design interaction surfaces needs to be aware of this danger [96].

– Voice interaction: Voice interfaces allow a hand's free usage of the system. Voice commands can be used to control the steps od the instructions of the AR system [167]. And as a response, the system can also narrate the instructions for the user [71]. Also, sound alerts are used to warn the user about some actions [167, 227]. Main issues are regarding noisy environments and also the use of headphones could be dangerous in some industrial scenarios.

– Haptic feedback: it is aimed to provide a stimulation of the sense of touch. Benefits have been shown regarding to increase performance [161], feeling of immersion [167], and add more dimension to the instruction presented by a visual AR [238].

Some constraints are regarding the portability of haptic devices. Thus wearable devices for large workspaces [161] and passive haptic feedback have been proposed [96].

– Wearables devices are not commonly used with AR systems. Nevertheless, a general purpose modular system in which wearable mouse and keyboard could help the user in case of changing configuration without the need to halt the operation have been proposed [26].

– Gesture based: Systems controlled by gestures, performed by the hands, aim to deliver a more natural interaction. It is worth to highlight that such strategies have to be robust regarding changes of skin pigmentation, bare hands or covered by gloves and noisy and grimy environments [70].

– Magic window: Place a display screen between the user and the environment that acts as a window with the virtual information added on top of the real background. Combined with a head tracking, this configuration could achieve a more natural result [14].

  Desk configuration has been proposed for the assembly [167] and also over multiple joint arms to improve user mobility [128]. However, this configuration is for relative static industrial scenes.

**Type of information** Considerations presenting the virtual information to the user regarding where and how.

The placement of the virtual information could be tricky, due to the users changing point of view and backgrounds, dimensionality variation of the devices, environmental factors and a large number of labels. Thus, virtual annotations should be projected accordingly to the user POV in order to be always visible [24, 69, 99].

For instance, the contrast between the background and the virtual element cannot be always guaranteed if the virtual object is always of the same color, given that it could encounter new objects with the same color [24]. Additionally, the connection between the virtual information and the real one can be lost if the real element is small.

Further, when placing text together with animations of actions required to perform by the user, the principle of spatial contiguity has been used in AR [71]. The spatial contiguity describes that information is more effective when is physically near to the animation [158].

Additionally, complex tasks require more information and the understanding of this information is limited by the user cognitive capacities. Also, understanding text requires more cognitive process than images, therefore less complex task should require less complex instructions [186].

Besides, the amount of information presented to the user should vary depending of the level of expertise for training systems. Thus, detailed in the early stages and gradually decrease [71].

## 2.6 Conclusions and Future Work

This study has been oriented to reach a general understanding of all the variables that could affect an AR implementation and to present some solutions already developed. Also, to propose to developers and researchers a global framework that could help to analyze future implementations by taking into account each one of the variables.

In this chapter, the characteristics of the elements of an industrial field (Domain Variables (DV)) that influence technical implementation of AR according to a study of state of art and previous implementations have been presented.

The clustering of the Domain Variables indicates that there are four main groups of influence related to the variables or characteristics of the user, the parts, the environment, and the task.

Similarly, it was found that these four groups have an influence over five main processes, two of them about how the AR system understand their surroundings and the user, another process related with the characteristics of the hardware and the system itself, a process about the strategies used in the interfaces, and, finally, the user perceptions and performance.

Each process has the same importance and should have the same relevance to the development. Systems should not focus only on the understanding of the parts and forget the user and the interfaces. Neither have strong interfaces but not taking into account the user performance.

Now, regarding the surrounding understanding, the main source of issues to be considered for any technique is that the systems encounter with many and very different "visual versions" of the same object. The reasons are due to the object is very different when is seen from different points of view, that is occluded, deformed, rusted, among others (Extended in Chapter 3).

Marker-based solution works in some situations such as "laboratory conditions" where it is possible to control the environment. But in real scenarios, it is not feasible to add markers to each part, especially, standard parts, and, in some other fields, they could change the perception of a user related to the part. Further, the markers are affected by light variations, grime, and only reveal a part of the visual information of the elements which is the relative position and orientation.

Moreover, there are two approaches for the use of the natural features of the elements to understand the surroundings. The first one is related to Feature Engineering, where the features used are predefined. The main withdraw of this technique is that it is time-consuming, and only works with the elements of this kind of features. Therefore, it is difficult to adapt it to new elements and new domain conditions that were not thought to be used.

The second one, Feature Learning techniques, lets the system to define the most discriminative features of the objects in a training phase. Thus, a large amount of training data, containing all the possible variations of the domain that the parts could encounter, is required. The methods for the acquisition of such training datasets still remain as a challenge for the reason that in many situations taking labeled photos is not feasible.

Summarizing, the selection of the used technique is situational, depending on how controlled are the conditions of the domain, how scalable for a new and large amount of elements the system needs to be, and how easy is to get training data. The presented DV in Chapter 3 could give a guide about the possible factors and how they influence the implementations.

Nevertheless, the current level of understanding of the surroundings is basic, limited only to deduce the characteristics, but not at abstract levels, such as reasoning about the surroundings and their implications, for instance, what it is happening and why.

A similar concept could be applied to the user understanding techniques, where variations to the user representations are due to skin pigmentation, lighting conditions, and user movements, among others. The use of external devices attached to the user is not always possible because of issues like social acceptance, and they could restrict the user mobility.

Visual-based techniques have to deal with high levels of occlusions and not limit the user to a restricted area. As well as in Surrounding Understanding, deeper levels of abstraction of the understanding of the user have not been reached. Issues such as being aware of the user actions, consequences and dangers remain unsolved.

With respect to the systems, the portability and commodity of the wearable devices still put on risk the acceptability of AR. Smaller, lighter wearable devices that do not hinder the user mobility are still required.

Additionally, greater accessibility to processing systems with higher capabilities will allow processing more complex and deeper scenarios, better understanding and human-computer connection.

Besides, little or no research was found taking into account the aesthetics or style of the devices used in AR, which could ease the user acceptability.

Regarding the user performance and perception, a balance needs to be taken into account regarding user immersion and user safety. In environments where there are a lot of movement, the user has to be conscientious about the external dangers. Better immersion leads to a better performance but also to less conscientious of the surroundings.

Further research is required about the user cognitive capacities and the interaction with AR and how to personalize the AR support depending on the user and the task leading to a better performance.

In relation to the user interfaces, the more senses are involved in an augmentation of the reality and more coherent the interactions lead to a better human-computer communication, better the experience is. The addition of haptics to a visual interaction increases the user immersion and performance, where a key element is a portability that allows the user to work in larger areas.

The main elements to take into account are the security while interacting, and the adaptability of the interfaces to changing environments and conditions of use. Further, new strategies for interaction that include physical elements of the environment with other senses are missing.

Finally, the key elements where the future research should focus are the ones that help to a deeper and abstract understanding of the user and surroundings, scalable and adaptable systems to changing conditions focused in the user commodity and mobility, and adaptability of the system to different tasks and users capacities to optimize the performance.

CHAPTER *3*

# Surrounding Understanding

## 3.1 Introduction

The Surrounding Understanding is composed of techniques that are aimed to interpret the environment and users surrounding. Many different processes are required to achieve an understanding of the real world. Some common process for performing this task are related with: segmentation, tracking, registration, sensing technique and features extraction.

In the next paragraphs will be presented the definition of some of the processes related with surrounding understanding as will be considered in this thesis. We use the definition of the main levels of task of object vision based on [13].

- *Detection* determine the presence of an instance in a stimulus
- *Localization* detect and return the position of the instance
- *Recognition* localize all the instances in a stimuli
- *Understanding* recognize and infer the role of the stimuli in its context.

An "instance" is defined as an occurrence of something that is of interest to the performed task. For example, for the task of object classification of industrial elements, an instance could be a type of class such as nuts, screws or bearings. More abstracts instances could described according to the proposed definition, if the task is related with human actions, example of instances could be walking or sleeping.

Some other support processes related are *Segmentation* is the process of dividing an image into its constituent parts and extract the one of interest. It has been considered one of the fundamental process given that usually is performed at the beginnings of the image analysis process affecting the subsequent activities [252].

*Tracking* is a segmenting method of a region of interest while keeping track of its motion and position. It solves the problem of the approximation of the path of an instance in an image while its move around the scene [16].

*Registration* is the process of finding the correspondence between two related images of the same instance. It is based on finding the geometric transformations (i.e affine) that allows to overlay the two images through objects features that remain invariant (i.e collinearity) to the transformations [57].

All of this processes relays on the identification of characteristics of the instances (*features*) that allows to differentiate them from the other instances and the environment. This *features* that can be previously defined (feature engineering) or automatically learned (feature learning) are the building blocks of surrounding understanding algorithms. Some example of features used in Object Recognition are: edges, vertex, color distributions, image moments, area, color gradients among others.

It is difficult to define the boundaries of these process in real implementations as one may contain or be required by another. For this reason here are treated as a general process and not specific algorithms. Additionally Artificial Intelligence (AI) and Machine Learning (ML) techniques have shaped the classical (feature engineer based Figure 1.4) structure of perform image analysis becoming more data oriented (feature learning).

In this section are presented the influence of the DV on technical implementations related to the processes intended to understand the surroundings. In table 3.1 are shown the founded subprocesses used for understanding the surroundings linked with the DV that have some influence on them.

## 3.2 Domain Variables (DV) Analysis

This analysis was made by first clustering the process that each one of the DV influences, and also defining what issues cause each one of them. Finally, similar issues caused by the DV were clustered (Figure 3.1). For instance, the presence of the user's hands, occlusions or incomplete parts, causes that the description of an object taken in real life being incomplete (incomplete measurement).

The results of the clustering of issues show that the DV influence the technical process in two ways. The first one is that they define *boundaries* in the characteristics of the technology, and the second is that they cause *issues* in the process of understanding the surrounding.

### 3.2.1 Boundaries or Domain Constraints

The elements of the domain that impose *boundaries* or constrain to the use of some techniques are: *a*) General appearance, *b*) knowledge a priori, *c*) working area size, *d*) parts dimensions.

For instance, depending on how important is the *general appearance* of the industrial elements, some techniques maybe are not possible to use. Marks or sensors used to easy recognition and tracking may generate the perception that the part is damaged, therefore, the use of markerless techniques is preferred [162].

Similarly, the possibility of having some knowledge of the environment such as 3D models, markers location or lighting colors define the available techniques to use. This

**Table 3.1:** *Domain Variables that affect each one of the subprocess of Surrounding Understanding according to authors.*

| Domain Variables | | Surrounding Understanding Process | | | | | |
|---|---|---|---|---|---|---|---|
| | | Object Recognition | Segmentation | Features | Sensing Technique | Registration | Tracking |
| User | Movement | | | ▨ | ▨ | ▨ | ▨ |
| | Velocity | | | | | ▨ | ▨ |
| | Hands | ▨ | | | | ▨ | ▨ |
| Environment | Knowledge | ▨ | | | ▨ | ▨ | ▨ |
| | Variability | ▨ | | | | ▨ | ▨ |
| | Env. conditions | ▨ | | | ▨ | ▨ | ▨ |
| | Ext. elements | ▨ | ▨ | | | ▨ | ▨ |
| | Area size | | | ▨ | ▨ | | ▨ |
| | Lighting | ▨ | ▨ | ▨ | | | ▨ |
| Task | Tools | ▨ | | | | | ▨ |
| | Errors | ▨ | | | | ▨ | ▨ |
| Parts | Gloss | ▨ | ▨ | ▨ | | ▨ | ▨ |
| | Color | ▨ | ▨ | | | | |
| | Texture | ▨ | | ▨ | | ▨ | ▨ |
| | General appearance | ▨ | ▨ | | | | ▨ |
| | Shape | ▨ | | ▨ | | ▨ | ▨ |
| | Occlusion | ▨ | ▨ | ▨ | | ▨ | ▨ |
| | Affine Trans. | ▨ | | | | ▨ | ▨ |
| | Gen. Dimensions | ▨ | | | | ▨ | ▨ |
| | Deformation | ▨ | | ▨ | | ▨ | ▨ |
| | Incomplete | ▨ | | ▨ | | ▨ | ▨ |
| | Arrangement | ▨ | ▨ | | | | |
| | No. parts | ▨ | | | | | |
| | Equal parts | ▨ | | | | | |

knowledge is commonly found in indoors or laboratory scenarios. On the contrary, in outdoors it is possible to have less infrastructure available and most of the techniques based on pre-installed elements cannot be used [26].

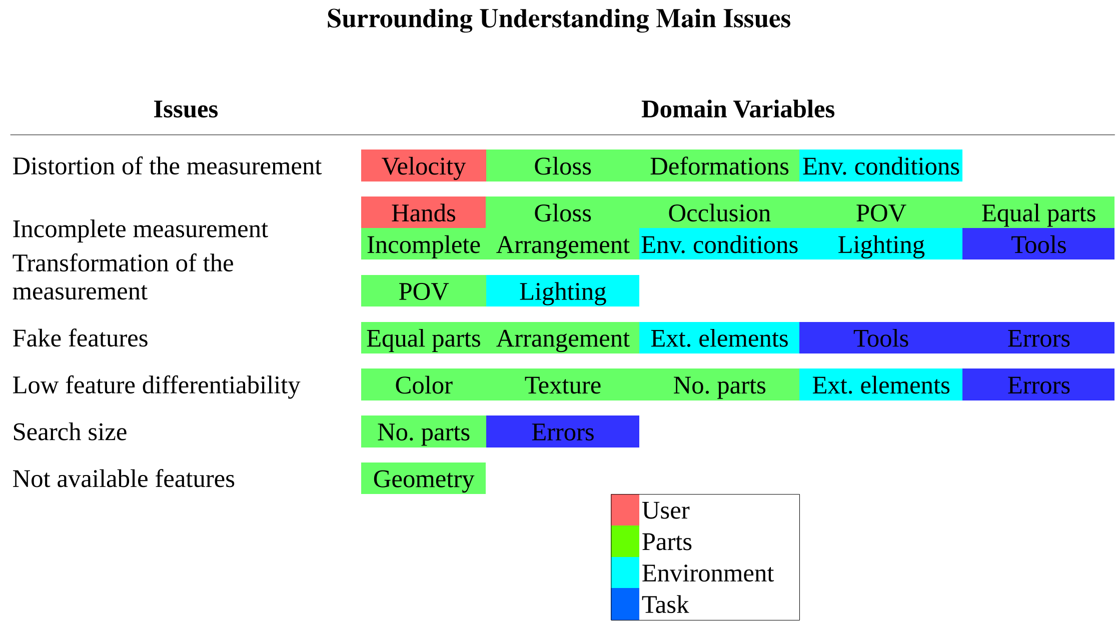
The size of the working area of the industrial operation also defines the typical distance at which the objects are viewed. This set constraints such as the features are large enough to be visible at the distance and the precision of the position of the objects. Large working areas allow the user to see objects from different distances which changes the available features [42].

Similarly, the size of the parts made or not possible to attach sensors or markers to them. Small parts are not feasible to use marker based solutions. Also, small parts cannot be easily recognized when are hold by the user, therefore, some techniques tend to fail [49].

### 3.2.2 Issues

The *issues* generated by the DV that affect the surrounding understanding process are presented in Figure 3.1. Here, we consider that the measurement is the description of an object.

The optimal measurement of the conditions of an object will be obtained by getting the image projection (Pinhole model) of the object in perfect conditions. As if it was taken in an empty space, with a homogeneous light coming from every direction, with no



Surrounding Understanding Main Issues

| Issues | Domain Variables | | | | |
|---|---|---|---|---|---|
| Distortion of the measurement | Velocity | Gloss | Deformations | Env. conditions | |
| Incomplete measurement | Hands | Gloss | Occlusion | POV | Equal parts |
| | Incomplete | Arrangement | Env. conditions | Lighting | Tools |
| Transformation of the measurement | POV | Lighting | | | |
| Fake features | Equal parts | Arrangement | Ext. elements | Tools | Errors |
| Low feature differentiability | Color | Texture | No. parts | Ext. elements | Errors |
| Search size | No. parts | Errors | | | |
| Not available features | Geometry | | | | |

🟥 User
🟩 Parts
🟦 Environment
■ Task

**Figure 3.1:** *Main issues generated by the DV (user-red, parts-green, environment-clear blue, task-dark blue)that influence Surrounding Understanding implementation in AR.*

variations or errors in its surface appearance.

Therefore, the main *issues* caused by the DV can be grouped into:

### Distortion of the Measurement

Considerable non homogeneously alteration of the optimal description of an object. For instance, rapid camera movements (due to user motion) produce motion blur which causes that techniques relying on corner detection to fail [129].

Similarly to the characteristics of the part, glossy or reflective surfaces can create false features by reflecting parts of the environment. Steven Henderson et al. reported that the use of passive markers illuminated by IR can cause fake reflections by metallic surfaces that were controlled by the camera exposure settings [97].

Additionally, visual techniques that compare the intensity of small regions of the image (patches) can produce erroneous matches dealing with glossy objects. Although, elements with few reflections can be handled by discarding the regions with a posterior process (e.g. pose estimation) or by simulating realistic reflections on training stages [143].

Another characteristic of the part that can distort the object's representation is the deformation or alteration of the objects shape. This alteration can affect techniques that rely on geometric proportions. For instance, to estimate the camera position matching 2D points present in an image with known 3D points, where it is usually assumed that the corners are rigid and that their internal spatial relationship does not change [250].

A possible solution for dealing with deformable objects is to model them as deformable meshes [182] and key points positions are defined by the weighted sum of the vertices in the model image, that changes when it is deformed [143].

Additionally, unfriendly environments typically found in the industry could affect the appearance of objects by adding grease, grime, dust, bad lighting, among others issues. Therefore in some cases, a marker-based solution may be not robust enough [72].

Likewise, electromagnetic and ultrasonic trackers may be affected by the interference from magnetic fields, metals and echoes [99].

#### Incomplete measurement

If we consider the measurement as a chunk of information that describes an object, some of this information can be missed compared to the optimal condition measurement.

This can be mainly due to occlusions caused by the user, highly reflective parts, other objects interposed between the target object and the camera, changes of POV, environment grime (e.g. dust or grease), extreme lighting.

Another source of lost information could be due to a physical loss of parts of the object or extreme deformation. In the next paragraphs, some common ways found in the literature of lost information and how the authors controlled this issue, will be described.

Industrial operations usually involve the user to physically interact with the parts for instance in an assembly operation. Thus, it is common that the user's hands cover the target parts that the object aims to recognize.

Therefore, recognition and tracking system is required to deal with small objects mostly occluded by the hand during operation. Damen et al. [49] propose an RGB-D system that jointly recognizes the hand and the handheld object. This approach works with objects large enough to be distinguishable when being held.

Similarly, fiducial markers attached to parts can be easily occluded and not easily attached to small parts. To overcome this limitation, Zhang et al. [250] suggest the use of IR enhanced computer vision by attaching reflective tape to the feature points of each part. The tape is not highly visible to human eyes and is only detected with IR cameras. Nevertheless, some incorrect render was reported when the user occluded some IR features of the objects [250].

Thus, one of the key points when dealing with occlusion caused by external elements with local approaches is that the non-occluded part contains enough information to be recognized [143]. For instance, the recognition system adopted by Hagbi et al. [87] use concavities as object descriptor and it works as long as enough concavities are visible.

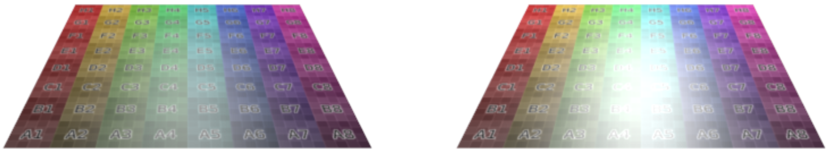
Further, the arrangement of the parts of an operation can be controlled in order to avoid parts overlapping or stacking among them [126, 167].

Moreover, another form of losing information is when there are important changes of the camera POV. In the case that relative position between the object and the camera is just rotating, the amount of information starts to decrease [87] until it reaches the point of having a completely different projective appearance, where new descriptors can be calculated [143].

On the other hand, when the camera moves from the target object, some details of the object are lost (too small). This issue can be solved by allowing to modify the model of the object online and fixing the number of features to force an update over the time [42].

Additionally, when the object comes in and comes out of the camera view, the system is forced to work with small regions of the object, where dynamic feature thresholds become weaker by decreasing the threshold by the low number of features and requiring several frames to increase it again [228].

Lighting can also generate lost in the description of an object, when the reflected light from the object's surface hides details of the object appearance. Figure 3.2 shows an example of a textured plane with a point light from the same point of view with different light energy. In the case (b) light saturates the object and some information related to the texture of the plane is lost.



**(a)** *Light energy 0.5*

**(b)** *Light energy 1.5*

**Figure 3.2:** *Representation of lighting effect on lost of object description. A textured plane with a light point with two different levels of light energy. In the case (a) Energy level is one unit more than (b). And the features related with the texture are lost*

This issue can be handled with local approaches, as mentioned before, where regions of the object that does not match are discarded in posterior process [143].

### Transformation over the measurement

Mainly two types of transformation were founded: geometric transformations given by the projection into the camera's plane with changes of POV, and, also changes in lighting conditions. Regarding to the first point, it always have been of great interest to develop system that are robust to the change of appearance of the objects when they are seen from different points of view, and, to recover or infer the object position and orientation in space relative with the camera.

Thus, conceptually, the main problem is the extreme change of appearance of the same object depending on the POV where it is seen. In literature, two common approaches were found.

The first one is to learn the typical view in which the object will appear but the features of the object will decrease once the angle of inclination of the camera reaches certain tilt, as it is shown in Figure 3.3 [86, 87].

The second one is to learn how the object looks from different POV. That can be obtained by having several different photos of the object under different positions [143], having projections (photos) from a viewing sphere around the object [47] or extracted from a 3D model [12].

Changes of lighting also transform the appearance of the description of the objects given different lighting intensities, colors or directions.

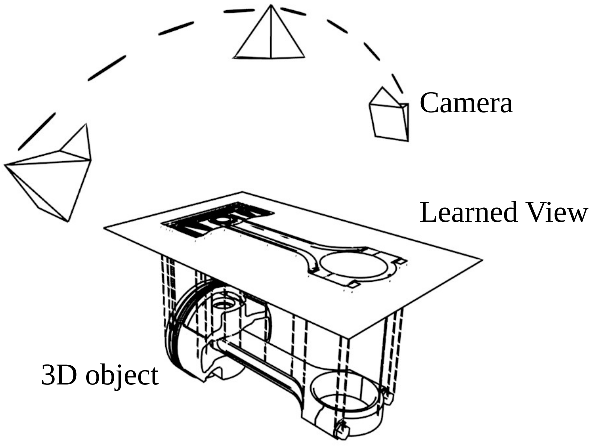


**Figure 3.3:** *Approach of learned view that represent a 3d object with camera at different tilts. As tilt increase less features are available.*

Several authors have mentioned the importance of descriptions that exhibit some robustness to lightning changes [143, 180]. For instance, tracking based on 3D models works independently of illumination and shadows conditions but requires the model of the object in its different configurations [42, 195].

On the other hand, controlled lighting setup has been also used in AR implementations [218]. The use of optical tracking systems, based on markers or light emitting sources, such as LEDs [99], requires controlled lighting.

#### Fake features

This issue is related to features that describe the objects that are similar to the target. This could be due to the presence of parts, tools or other external elements that exhibit patterns that are very similar to the target.

The simple scenario is that all the parts that the system will encounter have a unique shape, therefore, if there are several candidates that the system identify as the same, the one with better recognition score is selected [167]. Similarly, if there are parts that are exactly the same and can be used interchangeably.

Similarly, external tools used in the industrial operation could confuse the system if they have similar shape to the target objects [12].

Further, segmentation issues can be presented when objects with similar features are stacked or in overlapping. Additionally, the presence of objects with similar visual features in the background, such as those found in untextured elements, can be differentiated with the support of depth information [49].

Additionally, when dealing with assembly operations, many configurations or assembly states could look very similar in order to allow the system to identify when they are performed correctly and being aware of the errors.

#### Low feature differentiability

When the surroundings or other objects different to the target objects present characteristics very similar among them. For instance, when parts of the same class, such as different types of screws or washers, are in the industrial operation, only small details allow to differentiate them (intra-class).

Also, when the number of parts to be recognized increases, there is a chance that the features that allow differentiating them get closer hindering the object recognition (e.g [87]). A possible solution is presented by Jo and Kim [117] that consider only a small subset of objects depending on the place that the user is located.

Additionally, the number of external elements also increase the possibility of having objects with similar characteristics, that is common in industrial scenarios [12, 192]

Usually, in the applications it is searched to use features of the objects that present dimensionality disparity to facilitate their recognition and segmentation [105, 162, 167].

Nevertheless, applications in the industry usually present low color differentiability and lack of textures [49]. Thus, the use of local feature points and edges to generate descriptors have been used [12, 90]. Damen et al. [49] proposed the use of depth to support the discrimination of interest parts.

Similarly, lack of textures in the environment limits the use of techniques based on corner detectors,where a large number of this type of features is required, especially in tracking. Another alternative feature is the use of image intensity edges [129].

#### Search size

This issue is related to the number of parameters required to handle a large number of parts or variations of the objects. Given an increasing number of objects in the database, the time to recognize the objects and error rates grows rapidly due to the number of possible objects in the search space [117].

Further, an AR application that is used in many physical contexts encounters several objects in its use. Thus, a possible solution, as mentioned before, is to load the information only of the physical place where the user is located [117].

Another solution proposed by Ha et al. [86], for mobile implementations, is split the processes required in AR. On the server side it performs the OR based on bag-of-words, and, on the client side (mobile), it performs tracking and feature computation. Both parts are connected through Wi-Fi.

Additionally to the recognition, in some industrial applications is required to constantly keep tracking multiple objects including the user point of view at the same time. Henderson and Feiner [100] propose the use of two types of optical tracking, using reflective markers and infrared cameras. Although in many applications this setup is not feasible due the alteration of parts and controlled environment.

Besides the number of parts, considering the possible variations of the parts also increases the search size, for instance, checking assembly operations where each assembly step could be considered a new part. Solution based on the position of the objects has already been proposed [167, 195], but this type of implementations is based on a global coordinate system and is not fully aware of what was the error and how to return to a correct solution.

**Not available features**

Most of the survey implementations are tailored to specific industrial cases. For example, most of them are based on Feature Engineered techniques where the features that represent the object are explicit beforehand. This implies that future changes, such as the introduction of new parts, environment or lighting, could present a problem if their characteristics do not fit the previously defined system properties.

For instance, some of the more commonly found interest points (SIFT [147], ORB [189]) are available only in textured objects that are not commonly found in industrial elements [90]. Some others are constrained to objects with some characteristics, like the type of surfaces [12, 117], convexity [87], or the presence of some geometry invariants [167].

On contra proposition,Feature Learning techniques, such as Neural Network (NN), are able to learn the features of the objects and have shown to work almost with any type of object in any condition [94, 212]. But, one main concern is that they require a lot of labeled training data which in many situations is infeasible to get [45].

## 3.3 Object Recognition Methods

In this section it is presented how different proposed OR methods, over the pass years, behave regarding the most relevant Domain Variables (DV) found in literature of AR (Chapter 2). This survey is not exclusive for deep learning methods but a general overview of latest proposed OR methods.

The main driver of this section is the lack of structured knowledge about the different proposed approaches for performing OR, in the sense that, over the past years, several methods have been proposed, being conditioned to work or that only work under some restricted circumstances, such as requiring that the objects have some type of geometry or texture. Another known issue is that, usually, only successful cases are reported (under the conditions established by the method).

Besides, when large datasets are used, general measures can hide the failing cases. For instance, in the evaluation of the performance of a method with large datasets that contain objects with multiple characteristics, measures like Average Precision (AP) can hide the low performance in elements with certain characteristics (e.g untextured elements). Therefore, ablation studies should be performed, but in many cases, they require a lot of effort to obtain the labeled material (for each desired characteristic) and perform the necessary tests.

These issues make difficult to have a general map of what the limits of the current methods are. This make difficult to reuse or apply the methods in real life or to have a lot of experience in the field. Therefore, we propose to evaluate the approaches regarding how much they are invariant or consider the relevant DV.

Initially, 54 articles of OR related methods from 2017 - 2000 were collected, but only 31 articles, that potentially could be invariant to the next DV and relevant for industrial AR, were reviewed. Each DV is also described by the possible values or cases that it could take in order to evaluate the methods:

**Motion Blur** It is caused by the relative motion of the camera and the parts.
- Possible values: [static, moderate motion, fast motion].

**Gloss** It is related with the specular reflection of the parts. Therefore, the objects appearance is dependent of the view point given the reflection of the surroundings and the highlights produced by the lights.
-Possible values: [mate, fewer reflection, strong reflections, mirror like]

**Occlusion** It is the interference of other element and the camera's rays. Elements such as hands, same class element, tools, cut out of image (cropped), or other external element.
-Possible values: [non-occluded, partial-occlusion, nearly-full-occlusion, occluded by same class, occluded by other class, occluded by external element].

**Point Of View (POV)** These are the changes of the view point.
-Possible values: [translation, rotation, scaling, perspective, commonly viewed].

**Scalability** It is the behavior of OR with large number of parts and new cases (Generalization). Also, the ease of implementation and the effort for handling a large number of elements.
-Possible values: [minimal training data, synthetic data, multiple classes].

**Texture** It is the surface visual texture of the objects.
-Possible values: [strong textured (e.g patterns), textured, minimal textured, non-textured]

**Geometry** It is (In)Dependent of the geometry type.
-Possible values: [exclusive, required, exclude, any type]

**External elements** It is the presence of similar objects to targets objects in the background, multiple classes present in one image, or cluttered background.
-Possible values:[same class, other class, unknown, cluttered]

**Lighting** These are changes in lighting temperature, intensity or direction.
-Possible variations:[temperature, intensity, direction, multiple lights, single light]

**Errors** Variations in the objects such as deformations, incomplete or object variations.
-Possible variations:[surface appearance, geometric, intra-class variations]

### 3.3.1 Evaluation Methods

The evaluation of the OR methods has been made according to if the author expresses the influence of some variable, the characteristics of the testing database and the characteristics of the proposed methods. Besides, each method was evaluated regarding if it considers the possible values or cases that could take each DV with the next metric:

0. Not specified.
1. No present - minimum.
2. Low - fewer cases of the DV.
3. Medium - some cases of the DV.
4. Strong - most of the cases of the DV.

5. Reconstruct - it is able to recover the variable.

The methods were classified by the matching method (main method of the approach for finding the objects) and also by the type of used features. Table 3.2 presents the summary of the evaluation, where the maximum evaluation for each Domain Variable is picked from the methods with the same matching method.

| Matching method | Domain Variable Max. Value | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Motion Blur | Gloss | Occlusion | POV | Scalability | Texture | Shape | Ext. elements | Light | Errors |
| NN | 0 | 3 | 4 | 5 | 3 | 4 | 4 | 4 | 4 | 4 |
| Nearest Neighbor | 2 | 2 | 4 | 5 | 4 | 4 | 4 | 4 | 4 | 4 |
| Hierarchy Search | 0 | 3 | 3 | 5 | 4 | 4 | 4 | 4 | 3 | 1 |
| Boosting | 3 | 2 | 5 | 5 | 3 | 4 | 4 | 4 | 3 | 4 |
| SVM | 1 | 1 | 2 | 4 | 3 | 4 | 4 | 4 | 4 | 4 |
| Template Matching | 3 | 2 | 4 | 2 | 3 | 2 | 4 | 4 | 3 | 3 |
| CRF | 0 | 2 | 2 | 5 | 1 | 2 | 3 | 3 | 0 | 4 |
| MCMC | 0 | 2 | 4 | 5 | 1 | 2 | 3 | 4 | 3 | 4 |
| Minimization | 0 | 3 | 4 | 1 | 4 | 4 | 2 | 2 | 4 | 4 |
| Querying | 0 | 3 | 1 | 2 | 3 | 4 | 3 | 1 | 4 | 1 |

| | Score |
|---|---|
| 0 | Not specified |
| 1 | No present |
| 2 | Low |
| 3 | Medium |
| 4 | Strong |
| 5 | Reconstruct |

**Table 3.2:** *Summary of the OR methods evaluation, the score of the Domain Variable corresponds to the maximum score by the group of same matching methods for each Domain Variable*

As result, 10 main matching methods were found. The most common matching method together with Nearest Neighbor was Neural Network (NN) with 9 articles each one, followed by Hierarchy Search (5), Boosting (3), Support Vector Machine (SVM) (2), and one article for the others (Template Matching, Conditional Random Field (CRF), Markov chain Monte Carlo (MCMC), Function Minimization and Querying). The evaluation of all the reviewed articles is shown in Table 3.3.

The summary of the NN methods is shown in Table 3.4, where the most common used feature was raw pixels. And, only two methods use synthetic data as main input of the system [233, 235]. The most used datasets are: ImageNet [191], Pascal 3D+ [246] and CIFAR-10 [130].

Further, in the evaluation of the DV can be seen that glossy elements are merely considered with few elements in the evaluation datasets. As opposite, shape, lighting, erros and texture ar mostly covered by this type of technique.

Additionally, other issues are found in scalability by requiring large number of real labeled images, occlusion and external elements where most of the possible variations are not considered. Moreover, POV is mostly supported and some methods allow to reconstruct it from images [207, 235] both trained with 3D models.

In the Nearest Neighbor based methods, most common used features are interest-point based (SIFT, SURF, corners) followed by region based such as moments [155] and color histograms [39]. The datasets that were mostly used are proposed by their authors. And, in general, less methods considered untextured and full POV cases compared to NN. However, they have easier scalability compared to NN where some methods can be trained just with one sample [205] or not in all conditions [67]. These methods are presented in Table 3.5.

On the contrary, Hierarchy Search based methods (Table 3.6) did not consider the variations among the same class objects (errors), occlusions and glossy elements. But, they are robust in terms of lighting variations, external elements, shapes, textureless and scalability.

Three boosting methods were found, and they are presented in Table 3.7. It is worth noting that there is the only method that recover the occluded element and POV [245]. And, similarly to other methods, glossy elements were not considered.

Support Vector Machine (SVM) based methods are presented in Table 3.8 and other methods are presented in Table 3.9.

### 3.3.2 Conclusions

In general, it was found little attention to motion blur. One possible explanation is that it is not common and it is minimized using better cameras. Or, that most of the non-point based methods are robust to this issue. The only two datasets that possibly contemplate blur are the ones that contain elements in full motion such as KITTI [77] and UIUC [9] datasets.

The elements of most datasets only contains elements with few specularities or highlights, and, only few datasets proposed by authors contemplate elements such as metallic industrial parts [222] or methods working when minimal characteristics of the objects are available [48, 93, 248].

Further, POV was the most recovered property using methods such as NN [207, 235], CRF, MCMC, Boosting [245], Nearest Neighbor [82] and Hierarchy Search [48, 198]

One of the major known issues for some of the methods is that they require to have large amounts of labeled datasets that consider most of the variations that the target objects will be subject to. Additionally, for each characteristic to be reconstructed, it is required to be labeled. This increases the burden of getting datasets.

Further, the characteristics of the datasets have some bias that are usually not controlled in real life, for instance, the distribution of POV, that can influence the performance of the methods.

Regarding to the other DV (texture, shape, external elements, lighting and errors), there are support of almost all variations in most of the methods. However, there are still missing datasets and methods for creating them that allow the evaluation and study of concrete values that allow to isolate the influence of the variables.

**Table 3.3:** *Evaluation of reviewed Object Recognition methods regarding to the main Domain Variables of Industrial Augmented Reality. The methods are grouped by the main classification method. The evaluation score is as follows: (0) Not specified. (1) No present - minimum. (2) Low - fewer cases of the DV. (3) Medium - some cases of the DV. (4) Strong - most of the cases of the DV. (5) Reconstruct - it is able to recover the variable. Abbreviations: Occ: Occlusion, Scal: Scalability, Text: Texture, Ext: External Elements, Err: Errors*

| Method | Article | Blur | Gloss | Occ. | POV | Scal. | Text. | Shape | Ext. | Light | Err. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NN | [235] | 0 | 2 | 1 | 5 | 2 | 3 | 4 | 3 | 3 | 3 |
| NN | [233] | 0 | 2 | 3 | 3 | 3 | 4 | 4 | 3 | 3 | 4 |
| NN | [11] | 0 | 2 | 3 | 3 | 2 | 2 | 4 | 3 | 0 | 3 |
| NN | [210] | 0 | 3 | 4 | 3 | 1 | 4 | 3 | 3 | 3 | 4 |
| NN | [35] | 1 | 2 | 1 | 4 | 1 | 4 | 4 | 1 | 4 | 3 |
| NN | [247] | 0 | 1 | 2 | 2 | 1 | 1 | 3 | 2 | 2 | 4 |
| NN | [207] | 0 | 2 | 3 | 5 | 3 | 3 | 4 | 3 | 4 | 3 |
| NN | [155] | 0 | 2 | 1 | 3 | 3 | 4 | 4 | 1 | 4 | 1 |
| NN | [137] | 1 | 1 | 2 | 4 | 1 | 4 | 4 | 4 | 4 | 4 |
| CRF | [245] | 0 | 2 | 2 | 5 | 1 | 2 | 3 | 3 | 0 | 4 |
| MCMC | [245] | 0 | 2 | 4 | 5 | 1 | 2 | 3 | 4 | 3 | 4 |
| Nearest Neighbor | [208] | 0 | 1 | 2 | 2 | 3 | 1 | 3 | 2 | 3 | 1 |
| Nearest Neighbor | [82] | 0 | 1 | 2 | 5 | 3 | 1 | 4 | 2 | 3 | 1 |
| Nearest Neighbor | [155] | 0 | 2 | 1 | 3 | 3 | 4 | 4 | 1 | 4 | 1 |
| Nearest Neighbor | [39] | 0 | 2 | 3 | 2 | 3 | 3 | 4 | 3 | 2 | 2 |
| Nearest Neighbor | [67] | 2 | 1 | 3 | 4 | 3 | 1 | 4 | 3 | 1 | 3 |
| Nearest Neighbor | [137] | 1 | 1 | 2 | 4 | 1 | 4 | 4 | 4 | 4 | 4 |
| Nearest Neighbor | [27] | 0 | 2 | 2 | 4 | 4 | 4 | 4 | 2 | 3 | 3 |
| Nearest Neighbor | [58] | 0 | 2 | 2 | 2 | 3 | 1 | 4 | 3 | 3 | 3 |
| Nearest Neighbor | [205] | 2 | 2 | 4 | 1 | 4 | 2 | 1 | 3 | 4 | 1 |
| Boosting | [201] | 0 | 2 | 4 | 4 | 2 | 2 | 4 | 4 | 3 | 4 |
| Boosting | [149] | 0 | 1 | 3 | 2 | 3 | 4 | 4 | 3 | 2 | 1 |
| Boosting | [245] | 3 | 2 | 5 | 5 | 1 | 2 | 3 | 4 | 3 | 4 |
| Minimization | [248] | 0 | 3 | 4 | 1 | 4 | 4 | 2 | 2 | 4 | 4 |
| Querying | [93] | 0 | 3 | 1 | 2 | 3 | 4 | 3 | 1 | 4 | 1 |
| Hierarchy Search | [222] | 0 | 3 | 3 | 5 | 3 | 4 | 4 | 3 | 3 | 1 |
| Hierarchy Search | [219] | 0 | 1 | 2 | 2 | 3 | 4 | 4 | 2 | 3 | 1 |
| Hierarchy Search | [47] | 0 | 1 | 2 | 3 | 3 | 4 | 4 | 3 | 3 | 1 |
| Hierarchy Search | [48] | 0 | 3 | 2 | 5 | 4 | 4 | 3 | 4 | 3 | 1 |
| Hierarchy Search | [198] | 0 | 1 | 2 | 5 | 4 | 4 | 1 | 1 | 3 | 1 |
| SVM | [95] | 1 | 1 | 1 | 4 | 3 | 4 | 4 | 3 | 4 | 1 |
| SVM | [137] | 1 | 1 | 2 | 4 | 1 | 4 | 4 | 4 | 4 | 4 |
| Template Matching | [142] | 3 | 2 | 4 | 2 | 3 | 2 | 4 | 4 | 3 | 3 |

**Table 3.4:** *Summary of reviewed Neural Network matching method based proposals that potentially could be invariant or support main Domain Variables for Industrial Augmented Reality.*

| Year | Article | Features | Contribution | Experiments |
|---|---|---|---|---|
| 2017 | [235] | Pixels | Recognition on real images only based on 3DModels. | Experiments in real photos from ImageNet (PASCAL 3D+) databaseTrained with ShapeNet classification accuracy of 50.5% |
| 2017 | [233] | Pixels | Deep learning pipeline including data rendering, cost function modification and compact CNN architecture design aiming at objects recognition on real photos based on texture-less 3D model. | Accuracy of 47.2% with training with Pascal 3d and testing with real photos of ImageNet |
| 2017 | [11] | Pixels | CNN scheme for domain specific objection recognition tasks. Use minimal hardware resources ideal for low-end devices. | Experiment on CIFAR-10 accuracy of 81.46% |
| 2016 | [210] | Pixels | Weakly supervised CNN framework to reduce the effect of noisy labels due to data augmentation. | CIFAR-10 error: 5.11%, CIFAR-100 error: 26.42%, ILSVRC2015 20.78% error |
| 2016 | [35] | Pixels | Proposes a trace rule based self-organized map model built upon a sparse 2-stage deep belief network. That can generate more neurons with high SSI value which is beneficial to convey more useful and discriminative information for recognition. | Test 10 elements from ALOI dataset. Results show that trace learning rule is an effective way to associate current stimulus with historical activation from the same object and results in the development of transformation invariance learning. |
| 2016 | [247] | Pixels | Examine CNN architectures which are suitable for mobile implementation, and propose multi-scale network-in-networks (NIN) in which users can adjust the trade-off between recognition time and accuracy | Dataset: UEC-FOOD100: 75% accuracy |
| 2015 | [207] | Pixels | Scalable and overfit-resistant image synthesis (render-based) pipeline, with a CNN specifically tailored for the viewpoint estimation task | Dataset: PASCAL 3D+ |
| 2005 | [155] | Moments | Flexible recognition system that can compute the good features for high classification of 3-D real objects is investigated | Three texture-less sculptures of animals in different viewpoints |
| 2004 | [137] | Pixels | Assess the applicability of several learning methods for the problem of recognizing generic visual categories with invariance to pose, lighting, and surrounding clutter. | NORB dataset: 50 uniform-colored toys The objects were 10 instances of 5 generic categories with various amounts of variability and surrounding clutter, and lighting. Test error rates: Uniform background 7% error |

**Table 3.5:** *Summary of reviewed Nearest Neighbor based methods proposals found in literature that potentially could be invariant or support main Domain Variables for Industrial Augmented Reality.*

| Year | Article | Features | Contribution | Experiments |
|---|---|---|---|---|
| 2008 | [208] | SIFT | Method of OR and segmentation using SIFT and Graph Cuts. | 20 object models which are seen well in daily life and 100 test images. Model images were taken from the angles of every 45°. Test images were taken from various angles with various scales. Precision 1, Recall 0.81 |
| 2006 | [82] | SIFT | System for constructing 3D metric models from multiple images taken with an uncalibrated handheld camera, recognizing these models in new images, and precisely solving for object pose | Aligned a virtual square withan ARToolKit marker present in a modeled scene |
| 2005 | [155] | Moments | Recognition system that can compute the good features for high classification of 3D real objects is investigated | Three texture-less sculptures of animals in different viewpoints. Performance: 98·14% |
| 2015 | [39] | Color histogram | Method performs color image segmentation by a simplified pulse-coupled neural network (SPCNN) for the object model image and test image, and then conducts a region-based matching between them | Dataset: IJCV-data set: (Peformance: 41%). Their own dataset of 32 objects. Objects have between 10 and 40 possitive images. In total 783 positive experimental images, 71% correct matching |
| 2004 | [67] | Intensity-based invariant regions | OR approach which overcomes strong spite of viewpoint changes, occlusion and clutter and flexible objects | Own dataset of 9 model objects and 23 test images |
| 2004 | [137] | PCA – Pixels | Assess the applicability of several popular learning methods for the problem of recognizing generic visual categories with invariance to pose, lighting, and surrounding clutter. | NORB dataset: 50 uniform-colored toys The objects were 10 instances of 5 generic categories with various amounts of variability and surrounding clutter, and lightinig. Test error rates: Uniform background 13% error, high cluttered: impractical |
| 2002 | [27] | Shape sampled Points, Shape context | Approach to measure similarity between shapes and Use it for OR. The measurement of similarity is performed by 1. solving for correspondences between points on the two shapes, 2. using the correspondences to estimate an aligning transform. | Datasets: 2D objects the MNIST. 3D objects Columbia COIL: retrieval rate of 76.51 percent. |
| 2017 | [58] | Corners, Histogram of gradients. SURF | Use Graph-cuts as a segmentation technique. The segmented object is then recognized by mapping the feature descriptors of the images | ImageNet small subset |
| 2002 | [205] | Image points, Direction vector | OR system for industrial inspection. It uses similarity measures that are robust against occlusion, clutter, and nonlinear illumination change. Subpixel-accurate poses are obtained | 500 images of an integrated circuit under occlusions and clutter of various degrees. |

**Table 3.6:** *Summary of reviewed Hierarchy Search based methods proposals found in literature that potentially could be invariant or support main Domain Variables for Industrial Augmented Reality.*

| Year | Article | Features | Contribution | Experiments |
|---|---|---|---|---|
| 2009 | [222] | Vertex, Edges | Method for recognizing 3D objects and for determining their 3D pose. A model is trained solely based on the geometry information of a 3D CAD model of the object. They do not rely on texture or reflectance information of the object's surface, useful for industrial elements | Own dataset of real objects. Images of 8bit gray scale (640x480px). Two elemenst used a clamp and a fuse. 50 images of each object in different POV |
| 2013 | [219] | Edgelet | Implement method that they believe has some potential to be yet uncovered | Own dataset of 10 elements on clear background with partial occlusions. 76% accuracy |
| 2012 | [47] | Edgelet | OR method for rigid texture-less 3D objects for video input. The method is based on edgelet constellations with library lookup based on rotation and scale invariant descriptors | Dataset: ETHZ dataset (apple logo: 73.2; swan: 66.1; bottle 68.97; giraffe 72,4; mug: 60.9). The method is tested on a dataset of 30 texture-less objects: precision = 74% |
| 2011 | [48] | Edgelet | Generic, scalable and fast framework for concurrently searching multiple rigid textureless objects using edgelet constellations | Own dataset of 10 real tools sampled around viewing sphere |
| 2002 | [198] | Lines | address all the three major aspects of image registration: feature detection, correspondence and pose estimation | In one scenario a box is thrown on a table and the system correctly detects the object and computes its pose. |

**Table 3.7:** *Summary of reviewed Boosting based methods proposals found in literature that potentially could be invariant or support main Domain Variables for Industrial Augmented Reality.*

| Year | Article | Features | Contribution | Experiments |
|---|---|---|---|---|
| 2006 | [201] | Textons, Shape filters | Approach for learning a discriminative model of object classes, incorporating appearance, shape and context information. Given an image, the system automatically partition it into semantically meaningful areas each labeled with a specific object class | Own 21-class database, our algorithm achieves 70.5% |
| 2002 | [149] | Edge maps, intensity histogram | Show how maximum entropy framework can be used to combine simple discriminators | Testing: 24 images of each object (5 objects) under varying levels of occlusion, total 120 images. |
| 2016 | [245] | RGB luminance values, 3D shape, occlusion mask | OR pipeline that estimate multiple detected objects such as 3D pose, severely occluded by other objects, accurately estimating the occlusion boundaries | Dataset: KITTI: (6% AP, Pose estimation: 12%), Xiang and Savarese (2013) |

**Table 3.8:** *Summary of reviewed Support Vector Machine (SVM) based methods proposals found in literature that potentially could be invariant or support main Domain Variables for Industrial Augmented Reality.*

| Year | Article | Features | Contribution | Experiments |
|---|---|---|---|---|
| 2009 | [95] | Histograms of gradients | Pose-invariant object recognition systems using realistic 3d computer graphics models. provide a method for estimating the degree of difficulty of detecting an object | The first set of experiments dealt with the pose-invariant discrimination between two objects .EER 0.3%at 40,000 training samples per class. |
| 2004 | [137] | PCA – Pixels | Assess the applicability of several popular learning methods for the problem of recognizing generic visual categories with invariance to pose, lighting, and surrounding clutter. | NORB dataset: 50 uniform-colored toys The objects were 10 instances of 5 generic categories with various amounts of variability and surrounding clutter, and lighting. Test error rates: Uniform background 13%, high cluttered: impractical |

**Table 3.9:** *Summary of Conditional Random Field (CRF), Markov chain Monte Carlo (MCMC), Function Minimization, Querying, Template Matching based methods proposals found in literature that potentially could be invariant or support main Domain Variables for Industrial Augmented Reality.*

| Method | Year | Article | Features | Contribution | Experiments |
|---|---|---|---|---|---|
| CRF | 2016 | [245] | HOG | 3D Aspect Part representation: detecting object categories, determining their 3D poses and estimating the objects' 3D layout from a single image. | Datasets: - 3DObject (detection: 81.8 \| viewpoint: 80.7), VOC2006 Car (detection:48.7 \| viewpoint:85.9 ) and the EPFL Car (detection:97.5 \| viewpoint: 64.8 ), ImageNet (detection:90.4% \| viewpoint: 95.5%) |
| MCMC | 2016 | [245] | HOG | OR that handle occlusions from a 3D perspective. From a single image, is capable to detect objects, determine their 3D spatial layout and interpret which object occludes which | PASCAL VOC, LabelMe , ImageNet and our own photos |
| Minimization | 2012 | [248] | Probabilistic Shape | Select a sparse shapecombination from the dictionary that best represents the shape. Also to accurately segment the image taking into account the sparse shape combination and the image information | Own dataset of 10 real tools. |
| Querying | 2010 | [93] | Size, CIE LAB color, Pairwise Geometric Histogram (PGH) | Given an image of several objects on a structured background, they propose a segmentation and how features can be extracted for OR in mobile phones | Pill recognition, four different shape classes and 13 color. Highest recognition rate 100% |
| Template Matching | 2004 | [142] | Codebook, Spatial Probability Distribution | Automatically segments the object as a result of the categorization | Database: UIUC (Equal Error Rate (EER) of 91%) |

CHAPTER 4

# Physically Based Shading of Domain Variables for the Generation of Synthetic Datasets

In this chapter, a method for recreating relevant Domain Variables (DV) using a Physically Based Shading (PBS) approach is proposed, in order to create datasets for training and testing surrounding understanding algorithms. This method is framed under the industrial field, where the parts are very similar, present glossy effects and are subject to processes that change their visual appearance (e.g. corrosion or grime).

Building datasets is a complex activity that involves time and resources that are not aligned with the industrial world, that is in constant change and under considerable pressure by the market. Therefore, obtaining training datasets could be a problem especially when new products, procedures arrive constantly.

Synthetically generated datasets could be a solution to obtain almost free and fast training data, given that usually the parts and related information is already available by the companies. For instance, 3D models and visual appearance are known beforehand.

The problem with synthetically generated datasets is that usually, they underperform compared to photos based datasets because of the domain adaptation that is required to perform. Therefore in this research a method that blurs the difference between synthetic and real by simulating in a realistic manner the imperfections that usually occur in the industrial environment.

It is also known that although there are many available datasets with real photos, usually they are for general classes (i.e humans, cars, chairs...) as *ImageNet* dataset [191]. But in terms of parts of products produced by a company, there are limited options for acquiring training datasets.

In the case that a similar dataset with the same parts is found, it is required also to perform a *Domain Adaptation (DA)*, in order to fit the statistics of the dataset to the target domain. For instance, obtaining pixel-wise segmentation, where each pixel of an image is required to be defined to which class it belongs, or registration that requires defining the exact position and rotation of the object regarding the camera. Additionally, each time new parts are produced, it is required to obtain new photos and retrain the modes. Another option is to take the real photos, requiring a lot of resources and time.

Therefore, a method for creating synthetic datasets of industrial parts using solely the 3D model of the parts is proposed. The method allows to input the distribution of the characteristics of the real-life parts (i.e. lighting or pose) that influence the performance of surrounding understanding algorithms.

Further, current methods for creating synthetic datasets [114, 160, 177, 183, 214, 234] do not consider realistic shading or control the variations that are present in real life. And, both data augmentation and DA require to have a labeled or unlabeled dataset of the target objects.

Simulated variations are based on a previous study of the variables that influence the surrounding understanding algorithms in Industrial Augmented Reality (IAR) (Chapters 2 and 3). Therefore, the considered variations are: POV, intraclass variations, occlusion and external elements, appearance variations (surface roughness, grime, corrosion), no texture differentiability between classes[1], not shape or size restriction, lighting, and environment variations.

It is hypothesized that the proposed variables are the main source of the visual variations of an object in this context (industrial objects). It is worth to highlight that in this case there are not so common human-made textures where there will be infinite of patterns to recreate, but human variations are typically given by the geometry than usually serves a function and do not have large variations as could be in decorative or artistic elements.

Thus, natural variations of the materials are the main source of distortion of the "ideal" object representation such as corrosion or grime. The aim is to simulate realistic visual patterns from the interaction of light and relevant objects characteristics considering enough variations to avoid overfitting and to get closer to possible combinations of reality.

Therefore, the recognition system could use these patterns, or to be invariant to these variations, to identify visual patterns that could be useful in the recognition process.

This research proposes a set of variables that affect in larger scale the visual representation of objects, a method for physically based recreate these variables and automatically produce datasets for training algorithms used for surrounding understanding. Additionally, fully labeled data is produced where each one of the variations recreated is saved and could be used in training of another process such as segmentation or registration.

## 4.1 Introduction

One of the major goals of computer vision is to understand the world around us through images (2D projections of it). And it is a key component of other fields, such as robotics, Augmented Reality (AR) and automation.

Many different methods contribute to acquire and process the information coming from the images. High level process include image segmentation, Object Recognition (OR), objects tracking, 3D reconstruction among others. Where a wide variety of approaches have been proposed over the years for performing this processes [50].

[1] Metals are the only ones that corrode but material variation among the same class is also considered

Most recently, Machine Learning (ML) has been a powerful leverage for taking computer vision to real-world applications. Allowing to expand beyond of restricted domains to handle real-life complexity [196].

Further, Deep Learning methods have shown over the recent years a remarkable record-breaking in different challenging computer vision tasks. Those methods have shown an excellent performance not only in vision-related tasks such as segmentation, detection, classification but also in other pattern recognition areas such as speech and text processing [136, 178].

Deep Learning, by being a set of data-driven techniques, they adapt to a large number of perturbations (e.g. lighting conditions, occlusions, changes of Point Of View) that are present in the input data by finding relevant patterns regarding a given task [137, 253].

Nevertheless one of the most well-known drawbacks of having such versatility is that in most of the cases (supervised and semi-supervised) they require a considerable amount of labeled training data under a large number of variations [199].

Several alternatives have been proposed to decrease the dependence of manually labeled datasets and mitigate the efforts and resources of training deep learning models. To the best of the author's knowledge, there are four main approaches for dealing with this issue, Transfer Learning (TL), synthetic datasets, data augmentation and effective learning models.

TL is aimed to improve the learning of a predictive function (of a target task $T_t$) in a target domain($D_t$) using the knowledge of a source domain ($D_s$) and a source task ($T_s$) where the sources and targets are different ($T_t \neq T_s$ or $D_t \neq D_s$). Depending of these relationships, there are different TL settings [174] (Further explained in Section 5.1.1).

In visual applications in real-life is common to have the same task (e.g. detect the same class labels) but different domains for instance source of the images are different photos and paintings. This is referred as Domain Adaptation (DA), and it is a special case of TL [45].

Therefore DA and TL are aimed to take advantage of already labeled data from other sources and unlabeled data to reduce training and datasets creation efforts.

Moreover, a set of transformation can be applied to each one of the samples of a dataset to increase the number of variations, and this is known as *data augmentation*. Some techniques commonly used are geometric and color transformations, noise and blur addition, background blending and occlusions generation [10, 157, 254].

Besides, efficient models that can be trained with fewer and high data dimensionality have been proposed. This methods and models can handle strong overfitting allowing more effective generalization [113, 146, 229].

Another approach for easy the acquisition of labeled datasets for training Deep Neural Network (DNN) models is to digitally create them (synthetic datasets). By recreating the visual appearance and behavior of elements, external conditions, and variations that are of interest or affect the target task.

The use of synthetically generated data has been increasing over the years given the advances in computer graphics. Where some of the trends are related to the use of open source, gaming engines and realistic visual and behavioral simulations. Some of the most interest fields are features, scene and objects analysis in areas such as pedestrian, automotive and discrete products understanding [231].

Besides, the use of procedural created data has shown to be useful by their ability to provide large labeled datasets under controlled conditions. Allowing to systematically produce homogeneous training data, avoiding datasets bias (e.g. commonly viewed angles of objects) as usually are present in real life datasets [216].

Additionally, their inclusion in the learning process has proven to increase performance compared to the use of only real images [178, 207]. Also by controlling the different elements that could intervene in the performance of the target task (occlusions, POV, lighting among others) the synthetic datasets allows to make ablation studies for the understanding of the behavior of the systems [178, 214, 251].

Thus, the use of synthetic datasets for visual tasks can be considered a feasible alternative to transfer knowledge when there are scarce real labeled datasets.

Finally, there is also worth to notice that the use of these approaches is not excluding and that are commonly used together. For instance, perform a training with a synthetic dataset, augment the samples to avoid over computation and increase variations and perform a TL for fit the target domain statistics [176, 207].

Besides, one of the major concerns in the use of synthetically generated data is the influence of rendering technique and the fidelity in the reality recreation in the performance and learning process. Where previous work has demonstrated that increasing the realism improves the overall performance [178, 251].

Therefore, different implementations have tried to recreate photo-realistic datasets [118, 251] where some external variation are considered such as lighting direction and intensity, background and camera parameters. Nevertheless, realistic object's appearance variations have not fully considered (e.g Index Of Refraction (IOR), grime, corrosion, surface micro displacements).

In this chapter is presented a method for the generation of synthetic datasets recreating relevant domain variables, both own of the objects (e.g. IOR, corrosion) and external (e.g. lighting, occlusions) using a Physically Based Shading (PBS) approach. And it is analyzed their influence in the learning process of state-of-art of CNN models (MobileNet [106]) in an Object Recognition (OR) task.

Further, with the premise that the recreation of realistic patterns in images, generated through the simulation of physically based interactions of the light with objects and the variations that affect their visual appearance in real life, can be an useful information in the learning process of deep models.

A series experiments were performed training the CNN models with a synthetic dataset created with the proposed method and evaluated with real images. The synthetic dataset was created using 3D models and other required assets downloaded from online repositories and the synthetic data was automatically created from these using our method.

This chapter is organized as follows, in the next section,presented methods and studies for the generation of synthetic datasets (Section 4.2). Afterward, the teorethical background of Physically Based Shading (PBS) is presented (Section 4.3. The method for the creation of synthetic dataset and the process of simulation the Domain Variables is developed in Section 4.4. In the next section the recreation of the intrinsic variation of the objects is presented (Section 4.5). Similarly in the Section 4.6 the extrinsic variations are described. The environmental variables are presented in Section 4.7. Finally, discussion and future work is presented in Section 4.8.

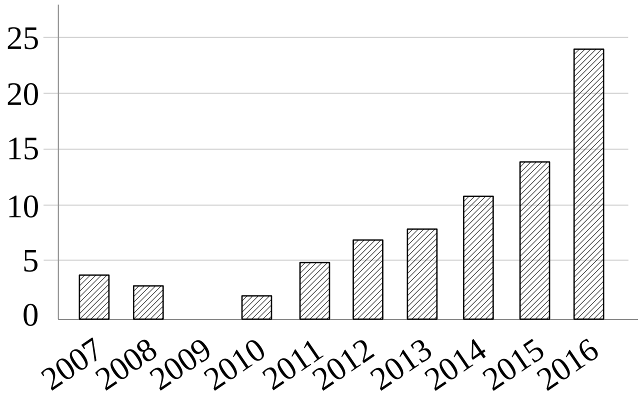


**Figure 4.1:** *Number of cited works by years related with synthetic datasets usage in [231]. Adapted from [231]*

## 4.2 State of the Art

In this section are presented methods and studies for the generation of synthetic datasets for training and evaluating *Surrounding Understanding* processes (e.g. Object Recognition, semantic segmentation, registration) are presented.

In 2017 Wang et al. surveyed the use of photo-realistic synthetic data for computer visual perception and understanding support, and coined the term parallel vision to the parallel interaction between synthetic and real data in vision problems. They found that there is an increasing interest in synthetic data (Figure 4.1) since the advances in computer graphics allows to create more realistic scenarios. And that their use is playing an important role in scientific research [231].

Additionally, they categorized the previous work under three main categories of the use of synthetic data: feature, object and scene analysis. Where three major trends were found, the first one is the increasing of realism due to the availability of powerful open-source and commercial simulation tools. The second is that synthetic data has permeated many computer vision tasks from low level (feature analysis) to high level (understanding). And the last one is the increase of interest and usage of synthetic data (Figure 4.1) [231].

In order to generate synthetic datasets, in 2017, Dosovitskiy et al. proposed a generative neural network for the creation of images of objects given a 3D model, object style, viewpoint and color [62].

Further, the study of the influence of realism in the creation of synthetic datasets has been studied. In 2017 Zhang et al. studied the influence of four rendering techniques, OpenGL with directional lights, OpenGL with point lights, Physically Based Shading (PBS) with outdoor lighting and PBS with outdoor and lighting objects such as lamps [251].

The synthetic data was used for three computer vision tasks, surface normal prediction, semantic segmentation and object boundary detection. The authors found that PBS with realistic lighting have a performance above the other methods and also for the three tasks an improvement over the state-of-art [251].

In 2009, Heisele et al. performed a study using realistic rendering of untextured objects for training a Support Vector Machine (SVM) pose-invariant OR system. The same objects were 3D printed and used of study the relation of training size and degree of detection difficulty of the objects. They found invariance respect the backgrounds but the accuracy was objects and view dependent [95].

Another source for obtaining realistic labeled data is from video games. In 2017, Johnson-Roberson et al. proposed a method for extract photorealistic data from a video game (GTA V) and used for automobile detection. Their results show that a state-of-art CNN architecture performs better with the synthetic data that with the manual labeled dataset [118].

Similarly, Richter et al. in 2016 presented an approach for creating pixel-wise semantic labeling from modern video games. One of the main challenges from obtained material from video games is that the internal content is inaccessible. They proposal consist of injecting a wrapper between the video game and the operative system that allows to control render commands of the game [188].

According to their experiments, they were able to produce labeled material corresponding to the manual approach used in [32] that would take 12 person-years in 49 hours. Also, they found that complementing real-life data with the one generated increase the accuracy on semantic segmentation [188]

Bochinski et al. in 2016 used the game engine game Garry's mod in order to generate different scenarios of the city traffic. Different "bots" were used to simulate the interaction between actors such as persons, cars, and animals with three different illumination setups (dawn, day and dusk). They prove that it is possible to simulate and detect real-world pedestrians, vehicles, and animals [31].

In 2017, Peng et al. proposed a Deep Generative Correlation Alignment Network in order to merge 3D model images with backgrounds and textures of the real domain in a realistic fashion. This approach aims to solve unrealistic match between foreground and background and the high contrast in the edges present in synthetic images. This approach shows to boost performance in pre-trained models on real image evaluation [176].

Another approach proposed by Tobin et al. in 2017, where instead of focus in achieving photorealistic results, they randomize different domain conditions such as distractor elements, texture, and characteristics of the lights [214].

They demonstrate that with a large amount of data random initialization present almost the same results as pre-trained models showing that is not necessary to generalize to the real world. Nevertheless, when less training data is used starting with a pre-trained model can significantly improve the performance [214].

Similarly, Peng et al. investigate how "low-level cues" such as realistic texture, pose and background affect the learning process of CNN. Particularly they investigate the difference between realistic texture (taken from photos and projected into the 3D models) and gray model rendering [177].

They found that when the trained model was fine-tuned for the task this low-level features are not required to be simulated. However, models without fine-tuning are not invariant to the cues. Additionally, in this study was not defined at up to what degree the model is invariant to object pose given that the only dominant poses were used [177].

Moreover, Massa et al. proposed in 2016 a method for adapting real images to 3D model view to select the 3d pose and model exemplar for a given real image. In order to avoid collecting labeled images, they used 3D models paired with their gray render view [152].

They hypothesized that given the features of real images is easier to deduce the features of a corresponding render point of view with a similar style and pose than "hallucinate" missing features in the real image details such as real object texture [152].

Sun et al. proposed an adaptation method based on decorrelated features for addressing the different bias of the synthetic and real datasets. They show that it is possible to use non-photo-realistic images that do not match the real images statistics by using the domain statistics. They showed that the difference between domain statistics leads to lower performance confirming that both source and target statistics matter [209].

Jiang et al. proposed in 2017 a pipeline for recreating configurable rooms layouts to generate photorealistic images using PBS and contextual relationship between objects, such as the relation between furniture. They shown that their approach reach a similar performance in different vision tasks (depth estimation, normal estimation, segmentation, 3D reconstruction and OR) than the NYU v2 dataset [163] using pre-trained models [116]

Another approach to reducing the burden of getting manual labeled material is to use efficient CNN models. For instance, in 2016, Wang et al. modified a CNN that use large amounts of synthetic data, with a multi-triplet cost function and a compact architecture in order to overcome the overfitting. The results in the classification task shown that the proposed CNN trained with only textureless 3D models is not lower that state-of-art CNN architectures trained with real photos [233].

In 2016, Movshovitz-Attias et al. studied the effect of the render parameters and realism in the creation of synthetic data on the performance in viewpoint estimation. They used state-of-art software (VRAY rendering engine) to render evenly distributed POV around cars 3D models and for each POV they explore different parameters (lighting, camera parameters, backgrounds, compression effects, among others) [160].

Regarding the render quality, they found that increasing the level of realism, the error is reduced, and that the error increase once the low quality renders are predominant in the train set. Additionally that the synthetic domain adaptation is not more difficult than the one required into real datasets. Further than combining synthetic images with a small set of real images improves the POV estimation [160].

Additionally, Aubry et al. in 2015 proposed the use of renders for analyzing the features generated by a CNN regarding domain factors such as object style, viewpoint, color among others [17].

They presented two type of "stimuli", 2D abstract shapes and 3D rendered views with a matte surface of their corresponding texture. They presented the different "stimuli" to a pre-trained CNN and analyzed the features response with a Principal Component Analysis (PCA) [17].

In another attempt for understanding what are the Neural Network (NN) learning by the use of synthetic images is presented by Pepik et al. in 2015. They explored which appearance factors are learned by the networks (POV, size, category and shape among others) together with different rendering styles and which ones can be improved by adding more data or there is required to have architectural changes [178].

Their results showed that state-of-art architectures are not invariant to some appearance factors (truncated, occluded and small objects). Also, that simply adding more data is not helpful requiring some architectural changes. Additionally, that detection improvement is proportional to the rendering realism when the synthetic data is combined with the real data. Nevertheless, synthetic data alone underperforms the real data performance [178].

Additionally, they found that the mix of synthetic data with real data, even in low-level of realism (wireframe), improve the performance of the trained models. And this improvement is proportional to the realism of the synthetic data [178].

Further, Busto et al. in 2015 proposed to improve coarse POV data labels in real datasets using synthetic data. They generated textured renders of cars with random backgrounds with defined steps in the viewpoint in spherical coordinates. And cluster the synthetic and real views for finding correspondences between them and mapping from the two domains. Finally, the labels on the real images are refined with a SVM trained with the transformed synthetic dataset [34].

Similarly, Su et al. presented in 2015 a pipeline for synthetic data generation for viewpoint estimation with a tailored CNN. The synthetic data generation instead of lookout for a realistic effect was focused on high variability in order to the CNN select robust patterns [207].

As result, they achieve better performance using millions of "renders" than state-of-art methods with real images. Where the accuracy was proportional to the number of synthetic images until reaching a plateau [207].

## 4.3 Theoretical Background - PBS

In this sections is presented the technical background and brief introduction for PBS the techniques used, the underlying physics and the description of the rendering problem.

Shading is the process of computing the color of objects of a 3D scene from a given Point Of View (POV). And it is the product of two things:

1. Light
2. Objects properties:
   (a) Geometric properties
   (b) Light-object interactions

The goal of photo-realistic rendering is to create images that are indistinguishable from a photograph from the same scene. Where Physically Based Shading (PBS) intents to model the interaction between light and matter based on physical laws.

### 4.3.1 Physics of light

Light is an electromagnetic radiation that is generated by oscillating charged particles at a given frequency. The visible light frequency lies between $4e + 14$ and $8e + 14\text{Hz}$. Light propagates both in the vacuum and in the matter. In matter, light is continuously absorbed and re-emitted.

Therefore, the interaction with matter depends on the material electromagnetic properties. The physical property that defines this interaction is the Index Of Refraction (IOR). It describes the ability of the matter to oscillate with the electromagnetic wave at a given frequency ($\omega$) [23].

The IOR of a medium is a complex number (Equation 4.2) which is real part is the velocity of light in space divided by the velocity of light in the medium.

$$n = \frac{c}{v} \tag{4.1}$$

And the complex part ($k$) that represents the attenuation of the light while passing through the medium.

$$\tilde{n}(\omega) = n + ik \tag{4.2}$$

In a micro scale, when the electromagnetic radiation enters the medium it shakes the medium's electrons. This originates an electromagnetic radiation at the same frequency than the incoming radiation with a phase delay. Then the net field is the sum of the incident and the radiated field of the electrons, where the last one masks the incident one [150].

Depending on the characteristics (phase, direction or frequency) of the resulting waves different possibilities could occur such as light scattering, reflection, transmission, or absorption. In light scattering the direction and intensity of light change in many directions while in absorption the electromagnetic radiation is transformed into another form.

### 4.3.2 Material-Light interaction

Most of the PBS approaches are based on a simpler model where light propagation is described using geometrical optics. Where light is modeled as a ray that travels in straight lines and instantaneously, and that the interacted objects are bigger than the wavelength of light.

Other assumptions made are that the optical systems are linear, energy conserving, there is no light polarization and the light environments are in a steady state. Finally, that light it can be emitted, reflected or transmitted [131, 181].

Therefore, most of the PBS models will focus on *light scattering*. Where there are typically found three reflectance mechanism, in Figure 4.2 are illustrated the three different lobes from the reflection of light from surfaces [244].

In the first case (a), the reflection comes from a smooth surface, that can be considered an infinite plane relative to the light waves between two different IOR mediums (Figure 4.3a).

Where the incident angle $\theta_i$ is equal to the reflected angle $\theta_o$ regarding the plane normal $N$. And the refracted angle is governed by the Snell law (Equation 4.3):

$$\frac{sin\theta_i}{sin\theta_t} = \frac{n_t}{n_i} \tag{4.3}$$

The reflection, in this case, is subject to the Fresnel theory, where light is attenuated depending on the incident angle $\theta_i$ and the IOR. Which main idea is that when the incident angle is close to the surface's normal the majority of light is transmitted, on contrast when the incident light is near to the grazing angle, the glossy reflection is the main source as is illustrated in Figure 4.4.

Thus, the Fresnel reflectance express the ratio between reflected and transmitted energy in function of incident angle, polarization and IOR. The ratio of reflected light is given

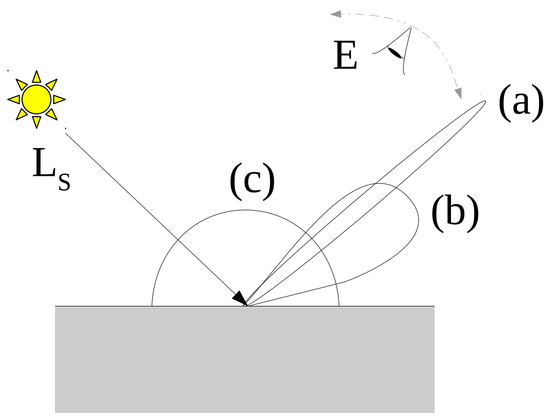
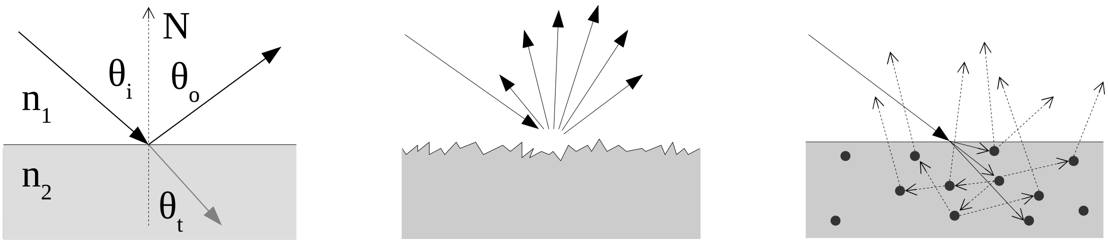


**Figure 4.2:** *Commonly found reflectance lobes of light from surface. (a) spike from smooth surfaces, (b) specular reflection from rough surfaces. (c) Diffuse lobe from subsurface scattering. Adapted from [244]*



**(a)**

**(b)**

**(c)**

**Figure 4.3:** *(a) Interface between media with different Index Of Refraction or ideal reflection. (b) Specular reflection on a rough surface. (c) Diffuse or body reflectance by subsurface scattering. Adapted from [244]*

by the Equation 4.4 that is the average of the polarized parallel $r_{\|}$ and perpendicular $r_{\perp}$ components of incident unpolarized light, where $0 \leq F_r \leq 1$:

$$F_r = \frac{1}{2}(r_{\|}^2 + r_{\perp}^2), \tag{4.4}$$

For dielectrics or isolators, the polarized components are given by the Fresnel equations 4.5 and 4.6:

$$r_{\|} = \frac{n_2 cos\theta_i - n_1 cos\theta_t}{n_2 cos\theta_i + n_1 cos\theta_t} \tag{4.5}$$

$$r_{\perp} = \frac{n_1 cos\theta_i - n_2 cos\theta_t}{n_1 cos\theta_i + n_2 cos\theta_t} \tag{4.6}$$

For metals or conductive material, the visible effect is not that perceptible given that

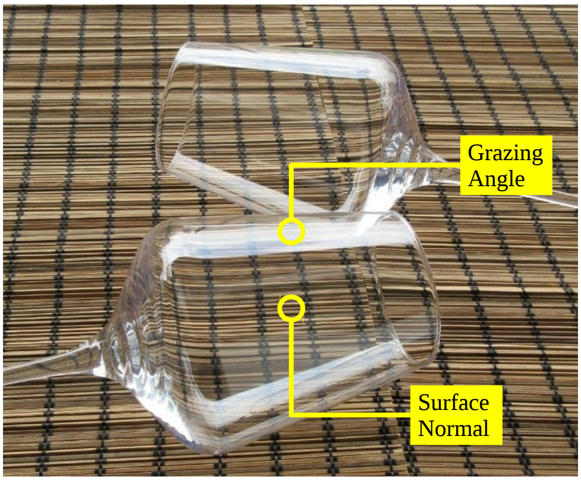


**Figure 4.4:** *Fresnel effect, incident light near to the surface normal is transmitted while near to its grazing angle is reflected*

they present high reflectance in all the angles. The parallel and perpendicular oscillation terms are expressed by equations 4.7 and 4.8 [135]:

$$r_{\parallel} = \left| \frac{cos\theta_i - (n+k)cos\theta_t}{cos\theta_i + (n+k)cos\theta_t} \right|^2 \tag{4.7}$$

$$r_{\perp} = \left| \frac{cos\theta_t - (n+k)cos\theta_i}{cos\theta_t + (n+k)cos\theta_i} \right|^2 \tag{4.8}$$

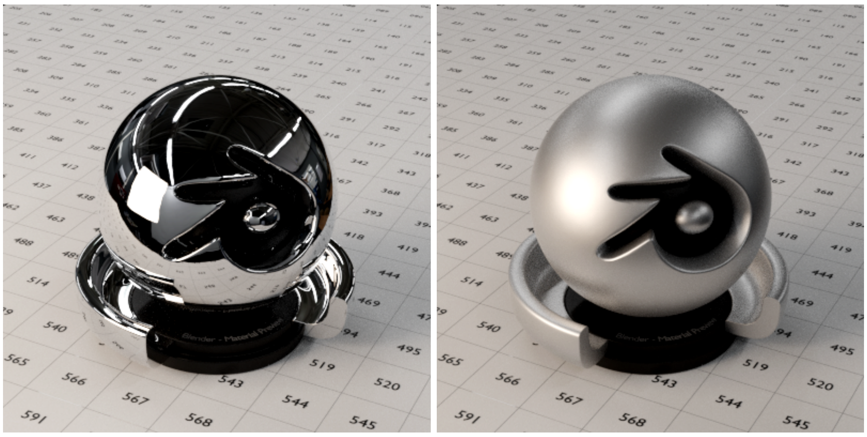
On contrary, body or diffuse reflectance (Figure 4.3c), is generated from subsurface scattering, where the light penetrates the material, is absorbed and scattered and finally leaves the material. In this interaction process light at different wavelengths is distinctively absorbed and scatter accordingly to the color of the material. As the bouncing of light in the scattering process tend to be infinite the direction of the output ray becomes random [92].

However, in the real world, few surfaces are perfect mirrors. Microscopic surface imperfections can be though as broken mirrors that behave as a collection of tiny flat surfaces. This is parametrized as *surface roughness* (Figure 4.3b) which produce a wider reflectance lobe (Figure 4.2).

These micromirrors are smaller than a pixel but larger than light wavelength, hence this imperfections can be seen or resolved. Usually, the roughness values of a surface go from 0 perfectly smooth surface to 1 that uniformly scatter light an all directions making blurrier reflections (Figure 4.5) [187].

It is worth notice that these micro bumps are at microscopic level and both surfaces as shown in Figure 4.5 appear equally smooth at the touch and visual senses.

The micro-variations in the geometry makes that each point in the surface reflects and refract light in different directions. The general appearance is therefore composed by the sum of the reflections and refractions. These micro variations of the surface at a



(a)

(b)

**Figure 4.5:** *Roughness render comparison. Same glossy material with low roughness (a) and high roughness value(b)*

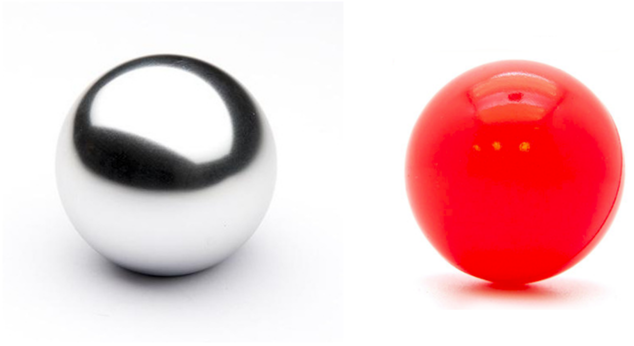
macroscopic level are treated statistically.

The Fresnel effect is less visible once the roughness of a surface increase due to normals of the surface diverge at a greater degree making blurrier reflections independent of the Point Of View influencing the apparent reflectivity on the surface (Figure 4.5).

### 4.3.3 Materials

Light is electromagnetic radiation, therefore, the optical properties are strongly related to the electric properties of the materials. Hence materials can be grouped into three categories: metals (conductors), dielectric (insulators) and semiconductors. In the real world, semiconductors surface are rarely visible. We will focus on metals and dielectrics (Figure 4.6).

Likewise, pure materials are unusually found, in real life materials have variations and impurities for instance metals can corrode and grime acting in cases as dielectrics. To achieve realistic looking materials there is required to consider these variations [193].



(a)

(b)

**Figure 4.6:** *Metal (taken from [2]) and dielectric (taken from [3]) examples*

In the previous Section (4.3.5), some BRDF were introduced, they describe the light

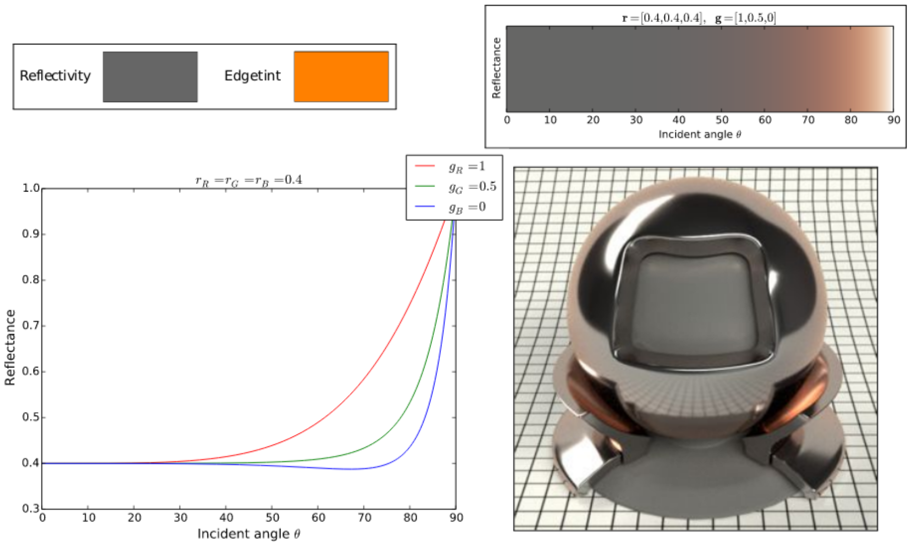


**Figure 4.7:** *In the left The Fresnel effect of a metal with different IOR values for the (R, G, B) wavelenght. At right the render illustrating the edgetint effect. Extracted from [85]*

scattering process of a surface at a particular point. The real-life materials are simulated by multiple low-level BRDF indicating which one use and their parameters.

### Metals

Metals or conductors are materials that have good electrical and thermal conductivity, thus the main optical response is due to the conduction of electrons.

The incident radiation in the range of visible light frequencies is absorbed in metals because of the available empty electron states. Allowing an electron to jump into a higher energy state, where the extra energy of the electron is equal to the energy of the photon. Similarly, the photon is re-emitted by the transition of the electron to a lower energy state. Thus the reflectivity of a metal is between $[0.90, 0.95]$ and only a smaller fraction of the energy is dissipated as heat [36].

Metals immediately absorb all the refracted visible light and re-emit at the same wavelength. When the light gets through conductive materials it becomes attenuated by the electrical conductivity with virtually none diffuse reflection. For this reason, the dielectrics are the only ones with diffuse reflections [244]. Therefore, the Fresnel effect is not that noticeable in metals as they are highly reflective from all angles.

Different metals absorb light depending on its wavelength, therefore they present tinted reflections that is governed primary due to the distribution of the frequencies of the reflected light and Fresnel effect blending the reflected color with a white reflection reaching grazing angles (Figure 4.7).

Their IOR depends strongly of the light wavelength an effect called *edgetint* illustrated in Figure 4.7. Where the reflected color is bias to one wavelength as it approach to white in the edge [85].

### Dielectrics

In dielectrics or isolators, the incoming light is both reflected at the surface (specular component) and refracted, scattered (diffuse component) and absorbed in some degree to then leave the material (Figure 4.8a).

The distribution of the incoming and outgoing light depends on the properties of the material. For shading, these distances are smaller than our sampling unity (pixel), hence we will assume that they are zero 4.8b

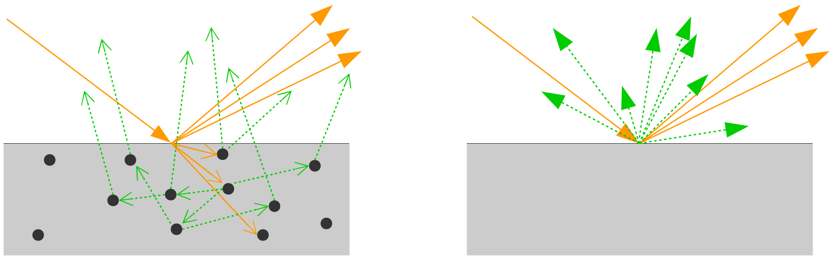


**Figure 4.8:** *Dielectric reflection with specular in yellow and body reflections in green. (a) theoretical reflection and (b) assumed light where the distribution of incoming and outgoing rays are smaller than the sampling unity*

The absorption of radiation in dielectrics is performed once the energy provided by the photon is greater than the *band gap* energy allowing the electron to leave the *valence band*. The photon is re-emitted by the transition of the electron to the *valence band* [36].

Thus the color is given by the band gaps of the material, where the input energy of the radiation is selectivity absorbed by the electron transition in the *valence band* and re-emitted in a different frequency.

### Blackbody

The radiation emitted by a body in thermal equilibrium with the surrounding electromagnetic radiation is determined by its temperature. This type of matter is called blackbody matter and the thermal radiation blackbody radiation. The law that describes the intensity of such radiation was derived by Plank in 1900 and its illustrated in Figure 4.9 [23].

The energy is emitted and absorbed in packages (*quantum*) that are multiples of the Planks constant $h$ which relates the energy $E$ and the frequency $v$ of a *quantum* $E = hv$. Thus at lower frequencies, many packages are emitted but their energy is minimal, and with higher frequencies, there is required more energy to produce the packages up to the point that it is not possible to produce more.

When the temperature of the blackbody decreases, the peak of its radiation curve moves to lower intensities and larger wavelengths as there is not enough energy to produce higher frequency packages. The average of the energy of the packages is known as the temperature of the body (Figure 4.9).

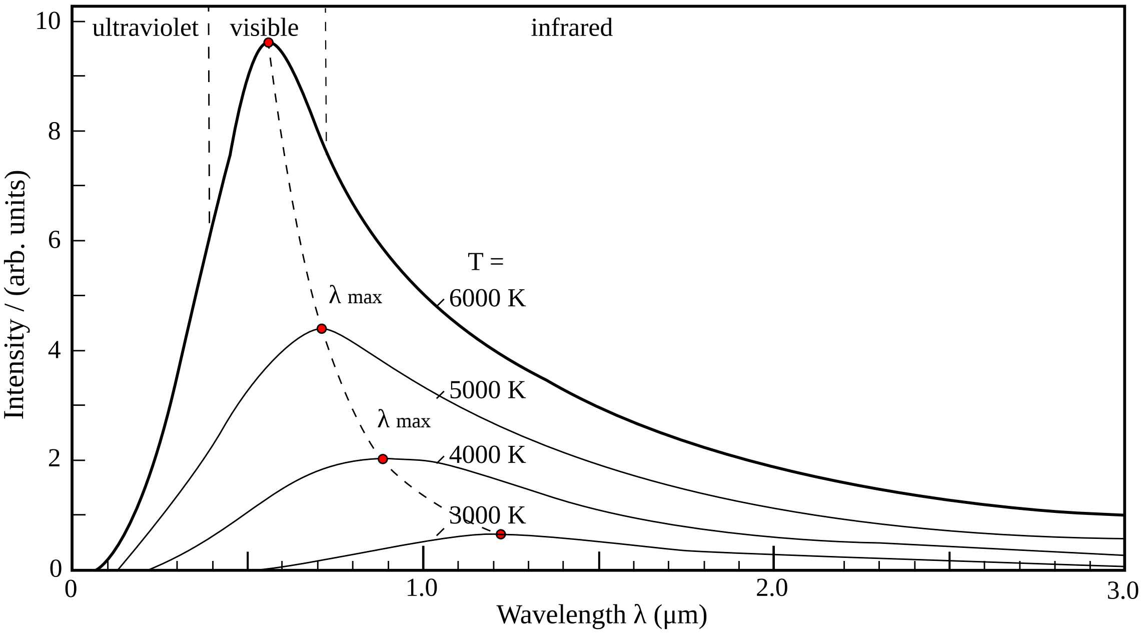


**Figure 4.9:** *Plank's blackbody radiation law. The backbody radiation curves changes depending of its temperature $T$. When the temperature decreases, the peak wavelength decrease moving to lower intensity and longer wavelengths.*

One of the main sources of visible light ($0.4 - 0.7\mu\text{m}$, Figure 4.9) is due to the emission of electromagnetic radiation of matter heated at high temperatures (Incandescence). The color of emitted light by a blackbody can be expressed in function of its temperature (*Color temperature (K)*). This relationship is represented in the *Planckian locus*.

The *Planckian locus* is the path of an incandescence blackbody in a *chromaticity space* as its temperature changes. The colors in the *Planckian locus* from $2500\text{K}$ to $20000\text{K}$ can be considered as white, where $2500\text{K}$ is reddish in contrast with $20000\text{K}$ blueish white (Figure 4.10) [46].

### 4.3.4 Radiometry

Radiometry studies the propagation of electromagnetic radiation and it's interaction with matter. The *radiance* one of the fundamental quantities of radiometry and it is a central quantity for PBS allowing to quantify and structure the light transport.

The *radiance* that can be though as the amount of Flux (energy) passing through an infinitely small solid angle hitting an infinitely small area.

*Radiance* is based on other radiometric quantities that are described in the next paragraphs.

**Radiant Flux** is the energy $Q$ per unit of time and has the unit Watt (Joule per second $1W = 1J \cdot s^-1$)

$$\Phi = \frac{dQ}{dt} \tag{4.9}$$

**Solid angle** defined by a point $P$ and a closed curve. Its magnitude is given by the radial projection of the closed curve onto a sphere ($A$) of unit radius ($R$). Each point of

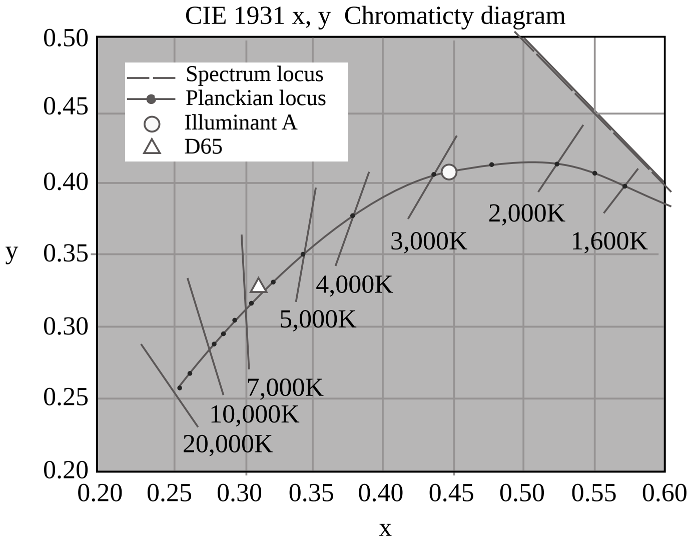


**Figure 4.10:** *Planckian locus in CIE 1931 chromaticity diagram. Taken from [46]*

the curve is projected to the point from which the solid angle $\omega$ is measured (Equation 4.10).

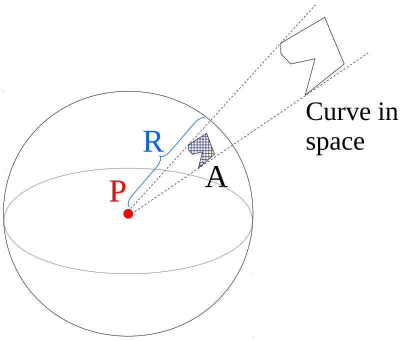


**Figure 4.11:** *Illustration of the definition of solid angle. Curve in space radially projected to the sphere in point $P$. Area of the projected curve $A$ is the magnitude of the solid angle*

The solid angle can be though as to how big an object is perceived by an observer from a given point. And it is calculated by projecting the silhouette of the object onto the surface of a unit sphere with center at the observing point (Figure 4.11).

$$\omega = \frac{A}{R^2} \tag{4.10}$$

**Radiant intensity** the amount of radiant Flux per unit of solid angle, passing, incident on, or emerging. It is described by equation 4.11:

$$I = \frac{d\Phi}{d\omega} \tag{4.11}$$

Thus, the **radiance** $L$, is defined in Equation 4.12 and in PBS it can be used to describe the behavior of a single ray of light.

$$L = \frac{d^2\Phi}{d\omega ds_o cos\theta} \tag{4.12}$$

Where, $\Phi$ is the radiant flux, $ds_o cos\theta$ is the projected area influenced by the light from the solid angle $\omega$, $\theta$ is the angle between the direction of the solid angle and the surface normal at the given point (Figure 4.12) [153].

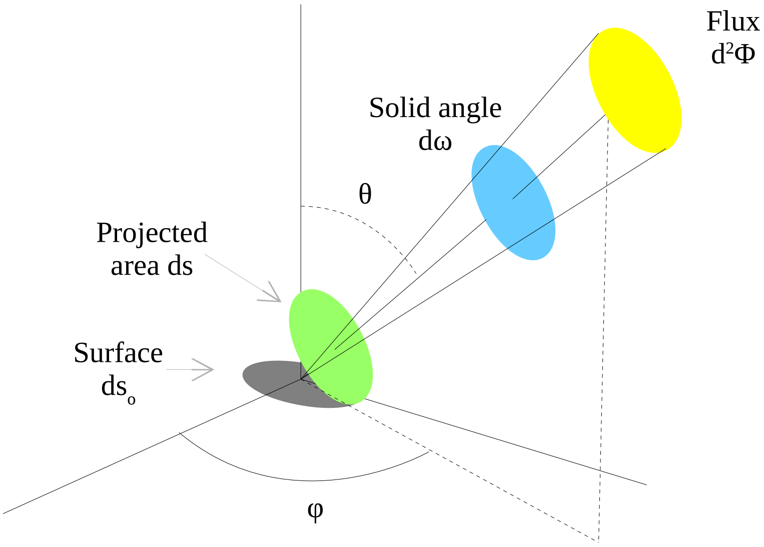


**Figure 4.12:** *Radiance geometric illustration. Adapted from [153]*

Finally, the **Irradiance** $E$, is the radiant flux per unit area arriving at the point $x$ from all directions in the hemispherical solid angle above the surface. And can be compute with the integral of the radiance $L(\theta, \phi)$ over the hemisphere $\Omega$, shown in Equation 4.13:

$$E = \int_{\Omega} L(\theta, \phi) cos\theta d\omega \tag{4.13}$$

### 4.3.5 Light Reflectance Models

The material properties of the objects define how they appear in different lighting and viewing conditions. In PBS systems the reflective behavior is described using a Bidirectional Reflectance Distribution Function (BRDF) that is a function that relates the incoming and outgoing radiance at a given point on a surface.

The sources of surface reflection models could be from [181]:

**Measured data** from real-world surfaces

**Phenomenological models** that mimic qualitative properties such as roughness.

**Simulation** based on the components of the surface

**Physical optics** using detailed models of light behavior computing solution to Maxwell's equations

**Geometric optics** models derived from known low-level scattering and geometric properties

The BRDF is a function that takes as inputs the incoming $\omega_i$ and outgoing $\omega_o$ ray directions and return the weighted contribution of the incoming ray to the final outgoing contribution. It define how much of the incoming light is returned towards the outgoing direction .The BRDF is formally defined in equation 4.14 [165]:

$$f_r(\omega_i, \omega_o) = \frac{dL_r(\omega_o)}{dE_i(\omega_i)} = \frac{dL_r(\omega_o)}{L_i(\omega_i) cos\theta_i d\omega_i} \tag{4.14}$$

The BRDF $f_r$ is defined in differential quantities because it relates only incoming and outgoing directions independently of other sources that might illuminate the surface. Where the outgoing radiance $L_r$ is weighted by the incoming irradiance $E_i$.

The last step is derived from the definition of irradiance in terms of the radiance (Equation 4.13). Finally multiplying both sides of equation 4.14 by the denominator and integrating over the hemisphere $\Omega$ regarding the incoming direction $\omega_i$ and is obtained the generic *reflectance equation* 4.15 for surfaces that are not emitting light:

$$L_r(\omega_o) = \int_\Omega L_i(\omega_i) f_r(\omega_i, \omega_o) cos\theta_i d\omega_i \tag{4.15}$$

This equation describes the output radiance $L_r$ in the $x$ point with direction $\omega_o$ as the integral over the hemisphere of the incoming radiance $L_i$ weighted by the Bidirectional Reflectance Distribution Function $f_r$, and by the Lambert cosine law, that express that the amount of light is proportional to the angle between the surface normal and the light direction $cos\theta_i = n \cdot \omega_i$.

Adding the emission radiance term, the *rendering equation* proposed by [120] is shown in equation 4.16. Almost all PBS models are an approximation of this equation which is completely based on physics and a standard in realistic computer graphics.

$$L_r(\omega_o) = L_e(\omega_o) + \int_\Omega L_i(\omega_i) f_r(\omega_i, \omega_o) |n \cdot \omega_i| d\omega_i \tag{4.16}$$

Physically plausible BRDF are positive and real valued $f_r > 0$, energy conserving, a surface cannot reflect than the incident light and it is reciprocal (Helmholtz principle) $fr_(\omega_i, \omega_o) = fr_(\omega_o, \omega_i)$ .

One of the main differentiability from different PBS models comes from the characteristics of its BRDF [187]. Some examples of famous analytical BRDF models are: Cook-Torrance [44], Blinn–Phong [30], Oren–Nayar [172].

### 4.3.6 Microfacets theory

Most of BRDF models are based on Microfacets theory which is an analysis of the effects of the microscopic surface imperfection in the reflectance. Microfacet theory was introduced to computer graphics Cook and Torrance [44] and since then many improvements have been made to this model.

The Cook and Torrance model BRDF is based on the addition of two components (Equation 4.17), the specular or surface reflectance $f_s$ and the diffuse or body reflectance $f_d$ (Figure 4.3).

$$f_r = sf_s + df_d \tag{4.17}$$

This two components are weighted such that $s + d = 1$. The diffuse reflectance is independent of the viewing point thus usually it is a constant. The specular component is modeled using the Microfacet theory, where the microscopic imperfections of the surface are view as microfacets, a small Fresnel mirrors, and the surface. As the microfacets are smaller than the sampling unit, it can be resolved, hence is treated statistically at a macroscopic level.

In the shading process of a sampling point on the surface, each of the microfacets on the point has a normal $m$ and share the viewing $v$ and the outgoing direction $l$. Given that the angle of the incoming and outgoing directions regarding the surface normal is equal to the specular reflection.

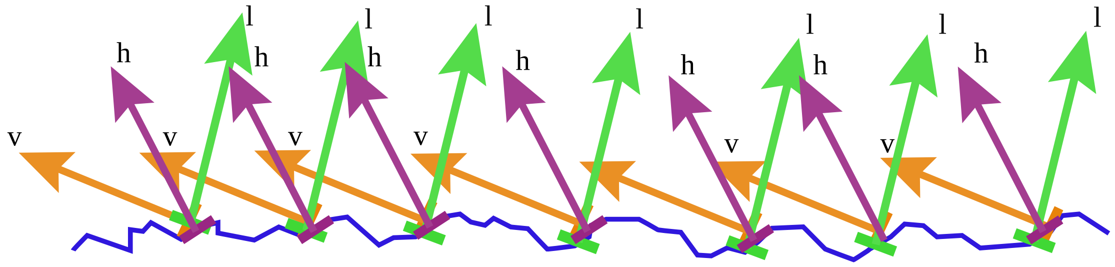


**Figure 4.13:** *Microfacets that contribute to the reflectance at shading point with incoming direction $v$ and outgoing direction $l$ are the ones that whose normal $m$ are equal to the vector $h$ (Equation 4.18).*

Thus the microfacets that contribute are the ones whose normal is equal to the vector that is in between the incoming and outgoing directions (Figure 4.13) Equation 4.18:

$$\vec{h} = \frac{\vec{l} + \vec{v}}{||\vec{l} + \vec{v}||} \tag{4.18}$$

Additionally, because of the microfacets are small structures, two important phenomena occur, the shadowing and the masking. The shadowing is the occlusion of light and masking is the occlusion of the viewing direction. Thus shadowed and masked surface does not contribute although there are inter-reflections usually they are not taken into account.

Therefore, the specular reflectance is parametrized by two statistical measures, the microfacets surface normals distribution (Normal Distribution Function (NDF) $D(h)$) and Shadowing-Masking Function $G(l, v, h)$.

The NDF $D(h)$ is the distribution function of normals evaluated at the active microfacet normal $h = m$. Indicates the concentration of surface whose normals are equal to $h$. This function defines the shape and intensity of the specular reflection. Usually, they are a function of roughness parameter with a Gaussian-like distribution. Once the roughness increases the concentration of microfacets around $n$ decrease [110].

On the other hand, the Shadowing-Masking Function $G(l, v, h)$ or the geometry term indicates the proportion of microfacets (with $h = m$) that are not occluded or masked. Thus, $G$ defines the probability of a microsurface under the conditions $(l, v, h)$ of being visible.

Finally the general form of the Cook and Torance specular term is presented in Equation 4.19, where $F$ is the Fresnel reflectance at the active microfacet:

$$f_r = \frac{F}{\pi} \frac{D(h)}{(n \cdot l)} \frac{G(l, v, h)}{(n \cdot v)} \tag{4.19}$$

### 4.3.7 Light Transport

The visual result of a scene is the result of infinite bouncing coming from different light sources and interacting with the objects present in the scene to finally arrive at the viewpoint. Each interaction with the objects is composed of a complex combination of reflection, transmission and light scattering. PBS requires the simulation of all this interaction known as *Global Illumination* (Figure 4.14).

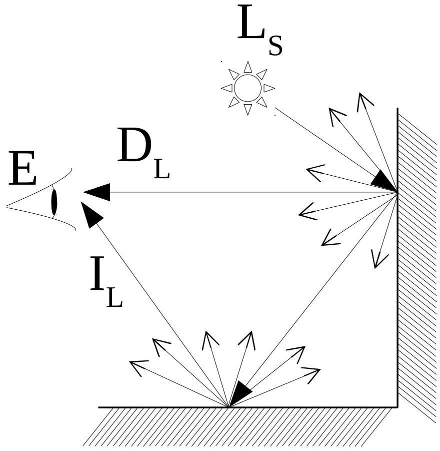


**Figure 4.14:** *Light rays are emitted from a light source ($L_S$) interacting with other objects. From this interaction the light rays sent back reaching the viewer eye or camera center (E) and are called Direct Light ($D_L$). Others rays interact with other objects before reaching the viewer (Indirect Light $I_L$)*

In essence, light that interacts once with a surface is called *direct lighting* and can be solved by solving the equation once, but the light that interacts multiple times with different surfaces, *indirect light* is required to solve the equation recursively. Global illumination involves solving both cases, direct and indirect lighting.

As presented in the previous Section 4.3.5, Kajiya in 1986 [120] introduced the *rendering equation* (4.16) that formally defines the global illumination problem. And presented a Monte Carlo algorithm called *Path tracing* in order to solve it. The main idea is to perform a back-tracing sampling of the Flux from the observer pixels until a light source gathering light from all light paths (Figure 4.15).

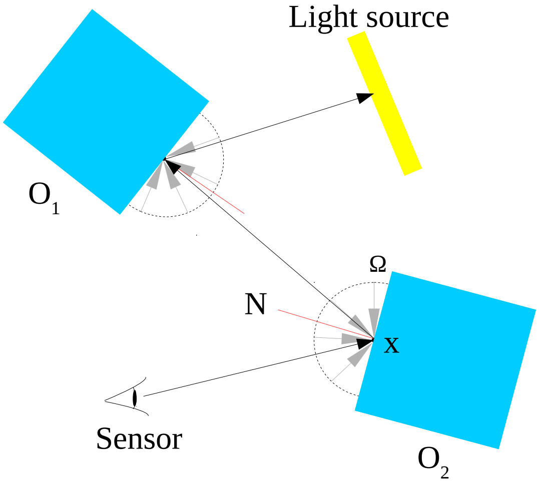


**Figure 4.15:** *Path tracing algorithm:*

Where *Monte Carlo integration* is used to estimate using random samples a deterministic integral ($I$) (equation 4.20):

$$I = \int_{\Omega} f(x)dx, \tag{4.20}$$

This is performed to sampling random $N$ points according to some density function $p$ and then computing the estimator (equation 4.21):

$$F_n = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)} \tag{4.21}$$

Therefore, to render the pixel's area $A$ composed by pixel's intensities $I_1, I_2 ... I_M$. With $W$ as the important function that describes how much of the arriving light from $\omega_i$ contribute to the output. The pixel measurement ($j = 1, ..., M$) is defined in equation 4.22 [131, 224]:

$$I_j = \int_A \int_{\Omega} W_j(\omega_i)L(\omega_i)|n_x \cdot \omega_i| d\omega_i dA(x) \tag{4.22}$$

This function run recursively as the light path bounce repeatedly between the scene surfaces. Where using the *Monte Carlo integration*, the pixel measurement $I_j$ can be evaluated using the estimator (equation 4.23) with some probability function $p(x)$:

$$\hat{I}_n = \frac{1}{N}\sum_{i=1}^{N}\frac{W_j(X_i)f(X_i)}{p(X_i)} \tag{4.23}$$

Since then, different variations of the algorithm such as Bi-directional path tracing [132] or Metropolis light transport [224] have been proposed intending to make a more efficient solution to the global illumination problem. Another type of approach for solving the rendering equation called *Radiosity* that is based on *finite element* making it viewer independent but requiring an expensive solution in complex scenes [80].

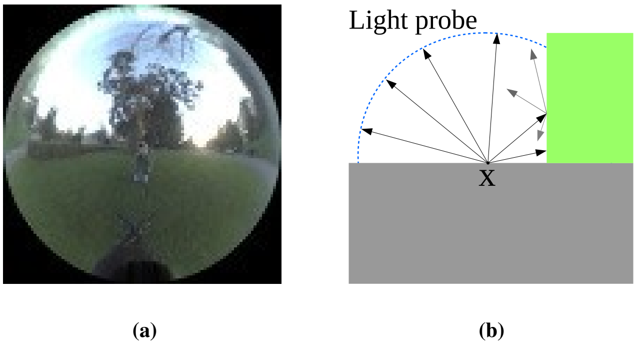
Path tracing is of unbiased nature, meaning that it does not introduce and any systematic error in the approximation of the rendering integral. The expected value of the estimator will be the mean of the population. This cannot be confused with consistency.

### 4.3.8 Image Based lighting

There are have been explored another alternatives less expensive such as *ambient light* or *Image-Based Lighting (IBL)*. In ambient light indirect lighting is simulated as a constant coming from everywhere giving unrealistic results.

On the other hand, IBL, Is the process of lighting virtual scenes with the light information from the real world. It uses real-world illumination captured in an omnidirectional High Dynamic Range (HDR). This *light probe image* has two properties, for every direction in the world there is a pixel whose values are proportional to the real-world light intensity [55].

Light probes are nowadays available on several websites as shown in figure 4.16a. Figure 4.16b is shown the interaction of the light prove with the radiance calculation at point $p$. The rays that are sampled from the hemisphere that does not encounter any object cast the values from the light prove. That is the equivalent to get the light values from the real environment.



(a)

(b)

**Figure 4.16:** *Probe light based lighting. In 4.16a and example of spherical light probe. In 4.16b Radiance interaction at point p with two surfaces. The rays sampled over the hemisphere takes the values from the light intensity of the light*

The representation of the environment in light-based methods could be considered as distant light containing the radiance information but not considering the BRDF material properties. [56]

IBL is a technique that let integrate virtual with real information that delivers realistic results without expensive calculations.

## 4.4 Physically Based Domain Variations

In this section is proposed a method for recreating relevant Domain Variables (DV) using a Physically Based Shading (PBS) approach.

The simulated variations are based on a previous study of the variables that influence the surrounding understanding algorithms in Industrial Augmented Reality (IAR) (Chapters 3).

Therefore, the considered variations are: POV, intraclass variations, occlusion and external elements, appearance variations (surface roughness, grime, corrosion), no texture differentiability between classes [2], not shape or size restriction, lighting, and environment variations.An example of the considered (and not) objects in this method is presented in Figure 4.17.

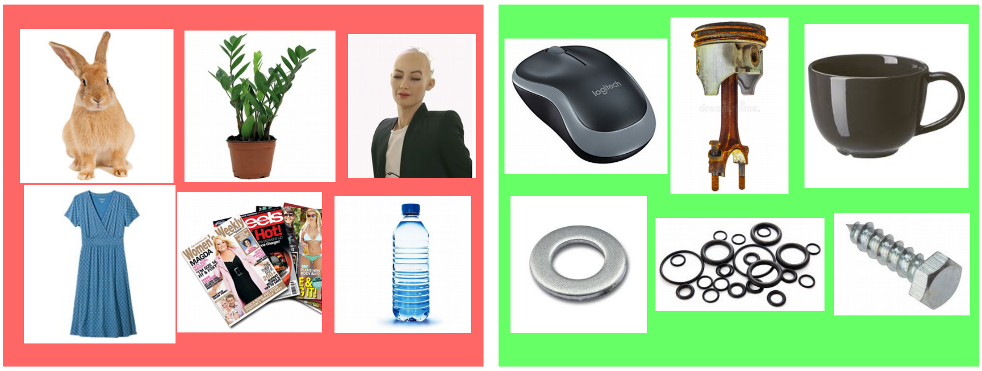


**Figure 4.17:** *On red example of not considered elements: plants, animals, humans, fabrics, textured objects (artistic), transparent. On green example of considered objects: metals and dielectric materials, complex an simple shapes, similar objects, glossy, with superficial variations, cluttered, multi-material objects.*

The domain variations proposed are shown in Figure 4.18 and can be classified into three groups. The intrinsic properties of the material and the environmental conditions that could be considered as general variations for many domains and the extrinsic variations that are not exclusive of the industrial domain but are the ones that most influence the change of appearance of this domain.

It is hypothesized that the proposed variables are the main source of the visual variations of an object in this context (industrial objects). It is worth to highlight that in this case there is not so common human-made decorative textures where there will be infinite of patterns to recreate.

But the variation of industrial objects are typically given by the geometry and the raw materials that usually serves a function and the sources of its variations are not as big as could be in decorative or artistic elements that are a function of the human imagination.

Thus natural variations of the materials are the main source of distortion of the "ideal" object representation such as corrosion or grime. The aim of this research is to simulate realistic visual patterns from the interaction of light and relevant objects characteristics considering enough variations to get closer to the possible characteristics of the reality.

[2] Metals are the only ones that corrode but material variation among the same class is also considered

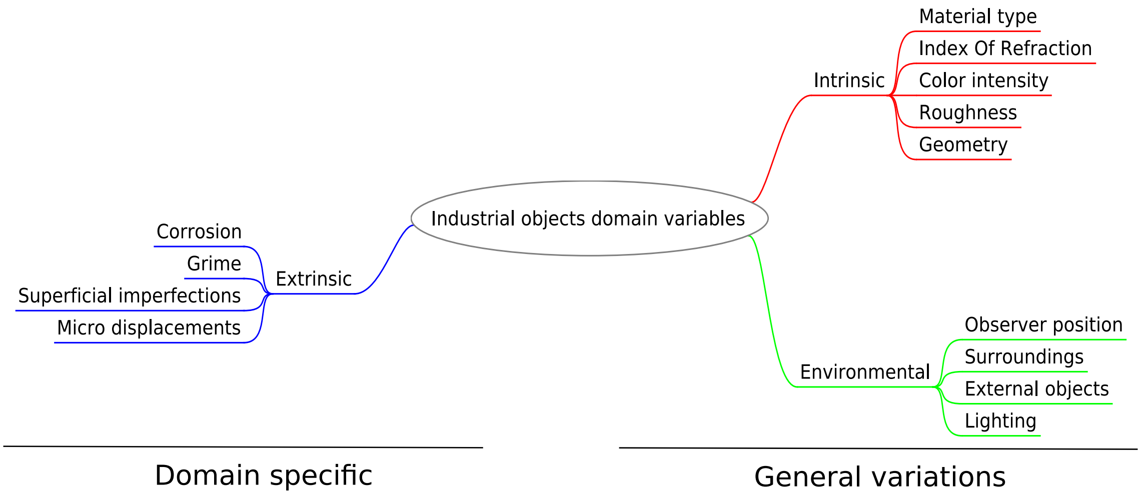


**Figure 4.18:** *Industrial domain variations considered. In general are the variations that could apply to other domains. The specific variations are proposed as main visual influence fro the industrial specific domain*

Therefore, the recognition system could use these patterns, or to be invariant to these variations, to identify visual patterns that could be useful in the recognition process.

In this research is proposed a set of variables that affect in larger scale the visual representation of objects, a method for physically based recreate these variables and produce an automatically datasets for training algorithms used for surrounding understanding. Additionally, fully labeled data is produced where each one of the variations recreated is saved and could be used in training of another process such as segmentation or registration.

This method could be applied to another type of elements where main variations could be identified and recreated, for instance in plants, variations such as diseases of shape deformations could be some of the main sources of visual variations.

### 4.4.1 Overview

The method propose an automatic form for generating fully labeled synthetic data with each one of the images containing the information regarding:

- Pixel-wise segmentation
  - Roughness
  - Index Of Refraction (IOR)
  - Color
  - Material type
- Domain Variables:
  - Surface micro-displacements type and amount
  - Surface imperfections type and amount
  - Grime type and amount
  - Corrosion type and amount

- Environment type and intensity
- Lights
  - Direction
  - Intensity
  - Temperature
- Main objection
  - 3D model
  - Position
  - Euler rotation

Further, the values of each one of the considered variations are drawn from a Probability Density Function (PDF). This helps to characterize in a more realistic way the form that the variations appear. At the same time having more control over the whole dataset ensuring the same distribution independently of its size.

The proposed method inputs are 3D models (with just the geometry, no UV mapping is required), and a set of 2D maps that define the possible distributions of some of the variations (e.g. grime, corrosion or environment).

With the previous assets, the first step is to prepare the 3D models which are performed automatically and it is described in Section 4.7.3. The next step is to render the samples which characteristics are defined in a configuration file in JSON format.

In this configuration file are defined the characteristics of the dataset such as renders for class, image resolution, sampling for each pixel, the processing device (CPU or GPU), number of samples per class or type of distribution and parameters for each one of the DV. An example of the configuration file is shown in Appendix 4.9.1

### 4.4.2 General Process

The main task of the generation of the samples consists of creating labeled images of a set of classes of objects. In summary, the proposed method consists of defining a number of Point Of View from where is the object class viewed. We have defined the POV as the position of the main object into space and their rotation in Euler angles thus the virtual camera remains static. For each POV the object class is rendered with different variations.

It is worth noting that one of the variations for POV is the geometry to achieve intra-class variation. Therefore for each sample generated for POV the 3D model corresponding to the class is also changed.

The general samples generation process can be seen in Figure 4.22 and it is composed by three main process:

i Setup: Load configuration file in JSON format and set the next parameters:

  - Rendering parameters: image resolution, sampling per pixel, processing device (CPU or GPU), tiles size at rendering time
  - Number of variations per POV and of POV per class
  - Paths of the materials, environments, 3D models and maps files

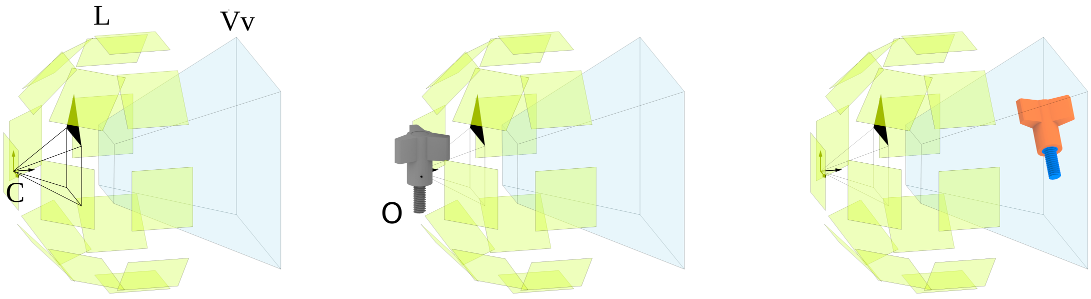
- Type and parameters of the distribution of the variations. Each one of the values of the DV such as light intensity, POV or material type, among others, at render time is drawn from a PDF specified in the configuration file. An example of configuration file is shown in Figure 4.9.1.

All the maps and 3D models are saved in separate files and in this step are loaded the available ones by listing the files in the path specified in the configuration file.

Additionally, are loaded from external files the base materials (Further described in Section 4.3.3) and the environments (Section 4.7. These process correspond to the (2) and (3) of the Figure 4.22.

After loading the main assets, a loop for each one of the classes start. For each of the classes, a set of different POV of the main objects is defined. And for each one of the POV, the samples are generated with different variations.

ii Loading objects: In this step the class and the external 3D models (defined as triangle meshes) are loaded from external files in OBJ format into the 3D space of the main scene. In the main scene are located the virtual camera and a setup of lights (Section 4.7). Additionally, both external and main object meshes are previously prepared given that they were downloaded from a web page (Process described in Section 4.7.3).



**(a)** *Default scene* **(b)** *Load main object* **(c)** *Position main object*

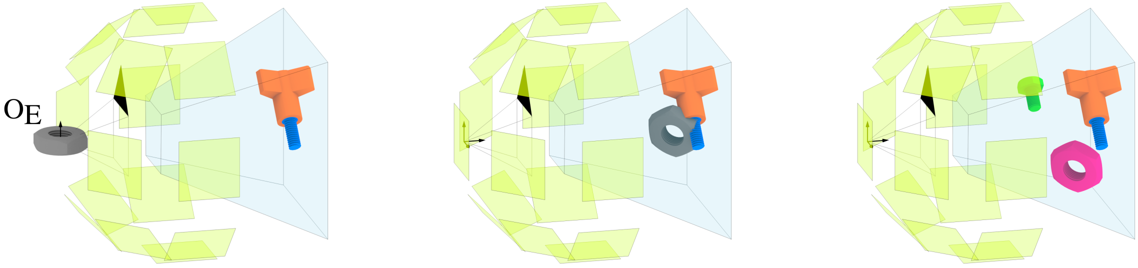
**Figure 4.19:** *(a) Default scene representation, contains the camera ($C$), and the lights set up ($L$) pointing all to one common point (lights center) that follows the center of the main object $O$. (b) 3D models are loaded from external files to the main scene aligned with global coordinates. (c) The main object is positioned in the visible volume (in blue $V_v$) and the materials are assigned to each one of the subparts)*

First, it is loaded the main-object model that is aligned with the coordinate system of the main scene. Therefore the object is located at $(0, 0, 0)$ of the main scene with no rotation. Then is positioned accordingly to the POV by setting object position and rotation. Notice that the camera and lights remain static with this approach and only the 3D models move (Figure 4.19).

After positioning the main object the material and its variations are defined and assigned (7 in Figure 4.22). The main object 3D model is also loaded with the information of which part of the mesh correspond potentially to a different material. Thus a loop for each one of the "subparts" is performed and defined the next parameters:

- Intrinsic:
  - Type of material: metal or dielectric[3].
  - IOR
  - Color intensity: In the case of dielectric material it refers to the color of the diffuse component. And in the metals, it refers to the tint of the reflections.
  - Roughness
- Extrinsic: defined by the type and amount
  - Corrosion (Only for metals)
  - Grime
  - Superficial imperfections
  - Micro-displacements

After defining all the materials of the main object, the process of loading external objects starts. They are meant to simulate occlusions, shadows and clutter in congruent consistently with the main object (same environment and light directions).



**(a)** *Load external element* **(b)** *Collision check* **(c)** *Multiple elemens*

**Figure 4.20:** *(a) Process of loading external elements ($O_e$) to main scene. (b) Collision check with other elements in scene. (c) Assign materials to external elements*

The process of loading external objects is fully explained in (Section 4.7.2). But in summary it consist Figure 4.20.

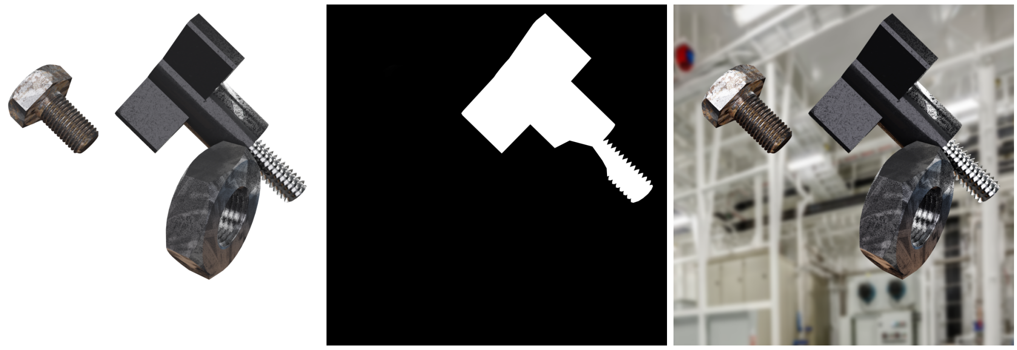
Each one of the parameters is drawn from a Probability Density Function (PDF) that were previously defined in the configuration file by specifying the type of the distribution (i.e. Gaussian or Exponential) and hyper-parameters such as mean or minimum or maximum range.

iii Generation: In this stage all the parameters are ready and the rendering process is performed. The main light (Light direction) is defined form the lights available in

[3] The limit of the values and type of some of the variations will depend on the material type. This is further explained in Materials Section (4.3.3)

the main scene (Figure 4.19a) is turned on and moved to point directly to the main object and similarly to the other variables its intensity is setup, and all the other lights in the scene remain hidden to the rendering process. Besides the virtual environment type and intensity are setups as well.

The rendering process is performed using the GPU and as result for each iteration the main outputs are, the synthetic image (Figure 4.21a), the mask (Figure 4.21b) that is an image of the same dimension of the render but in each pixel is specified with 1 if the pixel of the synthetic image corresponding to the main object or 0 otherwise (Figure 4.21).



**(a)** *Rendering* **(b)** *Mask generation* **(c)** *Augmentation*

**Figure 4.21:** *(a) Generating render according to properties of material, lights and camera (Section 4.3). (b) Pixel-wise segmentation of main object. (c) Data augmentation process, this is performed in real time during training (Section 4.7.4)*

As illustrated in Figure 4.21c a last process called *Data augmentation* require to complete the generation of the sample. In this step, a random background is added to the image as together with other transformations (Explained in Section 4.7.4). This step is performed during the training of the CNN algorithm in order to create more variation at the training time. Thus the rendered image is saved with the alpha channel to allow this operation later.

Finally, all the assets that do not belong to the main scene are removed to avoid an exponential consumption or resources, as it is an automatic process some of the 3D models contain up to 30 materials and for each one an instance is created.

This process was developed in a script in *Python 3* and used as main graphics library *Blender* API v2.78. *Blender* is a cross-platform open-source 3D animation suite. It completely supports the 3d pipeline including modeling, rigging, animation, and rendering. In addition, it allows employing Blender's API for Python scripting in order to write personalized tools. It has an embedded Python interpreter that can run scripts, draws the user interface and accesses to some internal tools.

The main reasons for used were that it is highly optimized for the render generation allowing full GPU integration for the rendering process. It supports to be used as a library from *Python* allowing the development of more complicated process (As the one presented here). It is free, open-source and allows to expand its functions if were required. Uses an unbiased rendering technique. More important it support the PBS implementa-

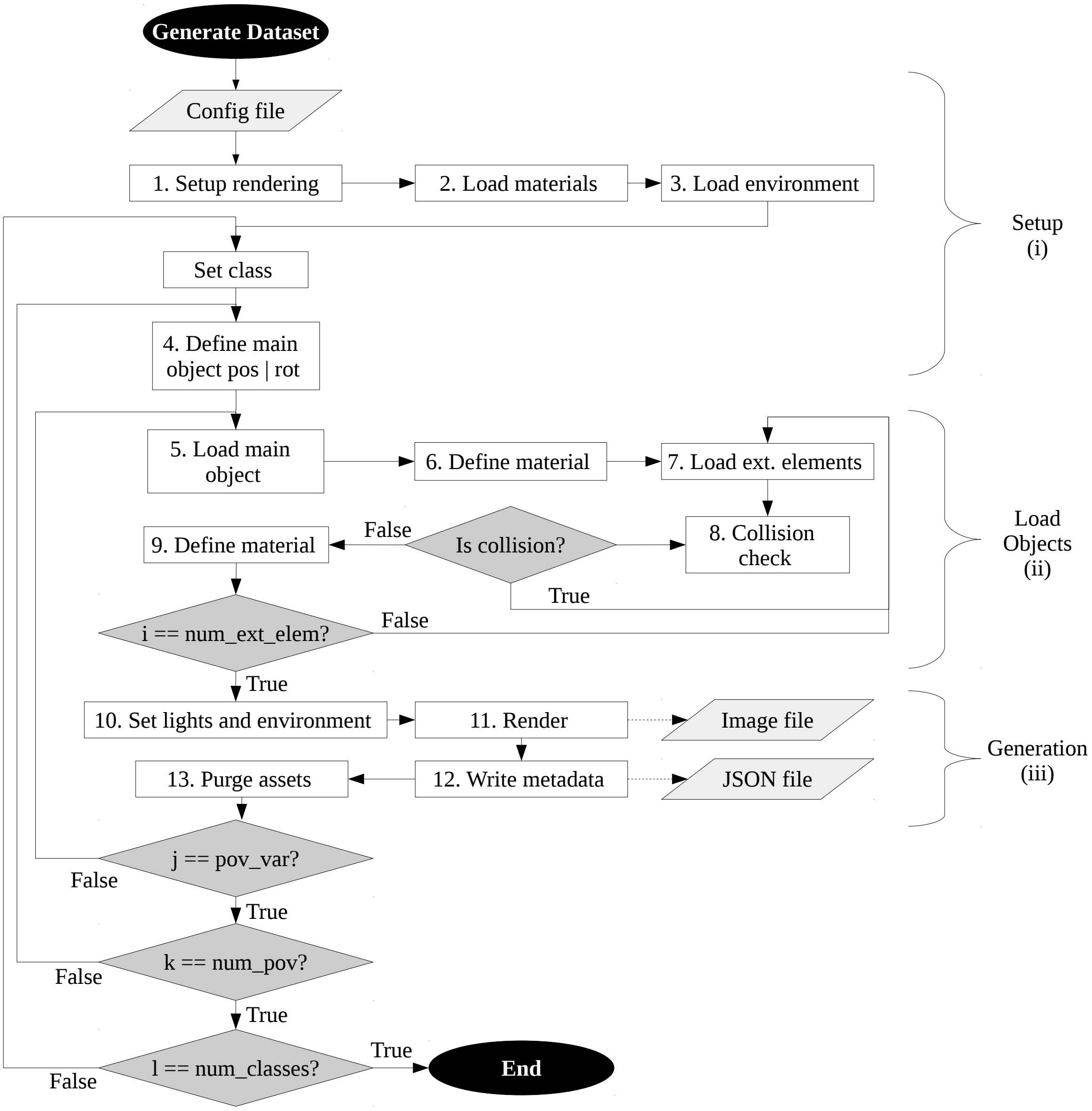


**Figure 4.22:** *General flowchart of the method for generating synthetic samples considering the DV with a Physically Based Shading approach. Performed in three main steps: (i) setup the type of distribution of the variables, path of the maps and rendering, (ii) loading objects (iii) rendering and saving meta data.*

tion as it have incorporated *Cycles* a *path tracing* rendering motor with already packed BRDF functions.

## 4.5 Intrinsic Variables

The intrinsic variables are related to the properties due to its own nature and not by its relationship with other elements. Together with geometry, the type of the material are the ones that properly defines an object.

The geometry variations are achieved through having different 3D models for the same class of objects. These models were download and prepared in a semi-automatic process explained in Section 4.7.3.

Accordingly to the types of material presented in Section 4.3.3, two main type of materials dielectrics and metals were simulated and used as building blocks for other variations such as corrosion.

Given that we decided to use Blender as our main rendering library, and in the moment of development (Blender version 2.78) was not implemented as default some characteristics such as Fresnel for metals and the interaction of Fresnel with roughness, there was required some tweaking for implement some characteristics.

The two basic materials were implemented based on [33,154,193] as will be presented below in this section. In Figure 4.23 were are shown the render of the two base material implementation in an environment used for visual testing of the materials. The virtual environment is composed of a surrounding lighting and the main light.

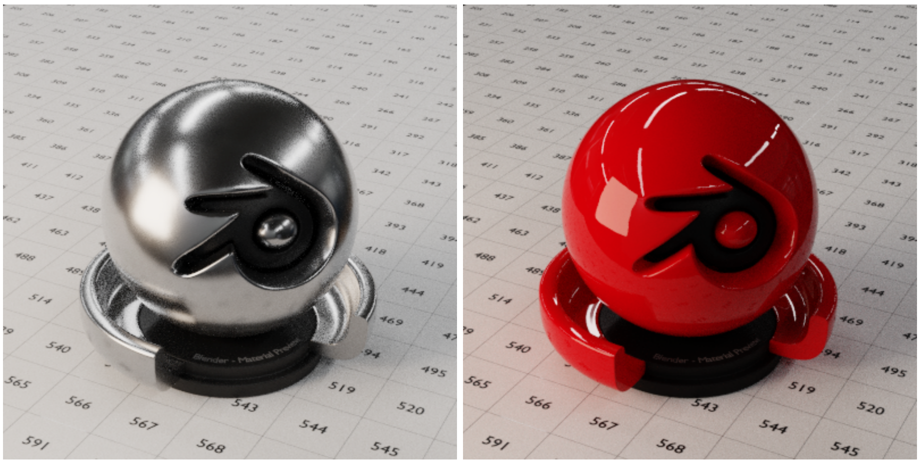
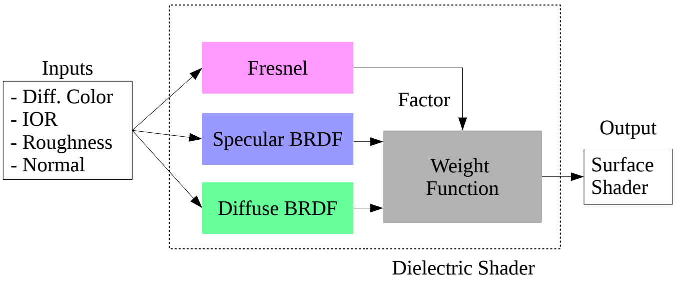

(a) (b)

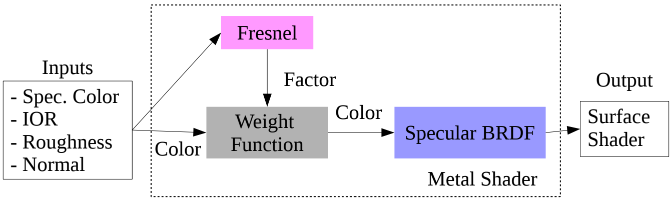
**Figure 4.23:** *Render of base materials, dielectric (b) and metal (a). Both were rendered in the test scene with environment and a main light direction for visually debug of the material's implementation. Metal variables: IOR: 1.45, roughness: 0.2, specular intensity: 0.4. Dielectric variables:*

### 4.5.1 Base Materials

**Dielectrics** The dielectric shader is the composition of a specular BRDF (GGX microfacets distribution[4] [230]) and a diffuse BRDF (Oren–Nayar [172]). This two functions are blended together according to the Fresnel effect (Figure 4.24a) using an weighted mean function (Equation 4.24).



**(a)**



**(b)**

**Figure 4.24:** *Diagram of basic materials. The inputs of both shaders are the same but the color in dielectric defines the diffuse and in metals the specular reflections. In (a) the Fresnel node defines the blending factor between the specular and diffuse functions. In (b) the Fresnel function defines the weight factor of the color mix of the specular reflections. The weight function used is a weighted mean operation*

The weighted mean operation is performed according to equation 4.24, where the result value $\vec{v_w}$, and the two inputs $\vec{v_1}, \vec{v_2}$ are weighted according to the factor $c$ :

$$\vec{v_w} = (\vec{v_1}c) + \vec{v_2}|c-1| \tag{4.24}$$

**Metals** the metal shader is composed by a single specular reflection BRDF [230]. And a Fresnel function that defines the factor between the base color of the metal and a white reflection in a weighted mean operation ((Equation 4.24)). The base color of the metal is defined by $(R, G, B)$ values (Figure 4.24b).

These materials are in agreement with the physical concepts presented in the previous section because of they do not break *energy conservation* as any surface can reflect more light than the incident, as long as the blending factor is between $[0, 1]$ (Figure 4.24a).

[4] Different models are available in Blender

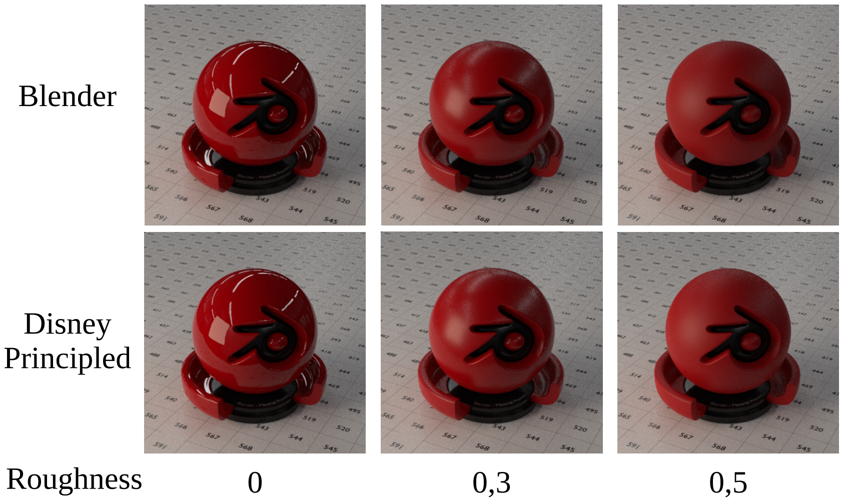


**Figure 4.25:** *Visual comparison between dielectric Disney principled material model (lower row) and developed materials. Both render in the same scene with same lighting. The IOR of the materials was* $1.45$ *and the specular value for the Disney shader was* $0.421$

### 4.5.2 Fresnel

As mentioned before the used version of Blender does not account for the effect of roughness in their Fresnel node. Blender already has implemented a physically based Fresnel node for dielectrics. The only issue is that it does not take into account the roughness of the material.

The Fresnel effect makes surfaces appear more reflective approaching to the grazing angle. This effect decreases proportionally to the roughness increase. We have implemented the workaround proposed by [193] to simulate this effect.

As the Fresnel effect is viewer dependent, The roughness value between $[0, 1]$ act as a factor between the weighted mean (Equation 4.24) of the surface normal and the incoming light direction($\omega_i$), the result is used as the normal for the default Fresnel node.

As the roughness increases the resulting normal tend to rotate towards the incoming direction. This effect makes that when roughness is set two one, the Fresnel effect is set as if the point were looked from the incident angle making the Fresnel effect to disappear. Additionally as in [33] the values of the roughness were mapping (squaring the input value) to have a linear perception [193].

Although this is a rough approximation given the absence of Fresnel for metals. Where input values are not physically but artistic descriptive. In our case, it is not relevant as our approach simulates possible configuration of parameters, and it is not an experimental simulation. Therefore while the parameters are in a range of possible values they will produce physically viable renders, and the result will be the same as used physically correct parameters.

As comparison, in Figure 4.25 are shown the comparison between the Blender proposed materials (Top row) and PBS Disney principled model [33] (Lower row) at different roughness.

The specular value ($s$) equivalent from the two compared materials in the dielectric materials is calculated using the equation for special case of the fresnel:

$$s = \frac{1}{0.08}\left(\frac{n-1}{n+1}\right)^2 \tag{4.25}$$

The other parameters of the intrinsic variables are drawn from a range of possible values in real-life and are set to the base materials (IOR, roughness, color(specular for metals and diffuse for dielectrics)).

## 4.6 Extrinsic Variables

The extrinsic variables are given by the relationship with a foreign process and not properly of the objects.

### 4.6.1 Corrosion

Corrosion is a chemical or electrochemical reaction of a metal (usually) and its environment that produces the deterioration of the properties and the material. It is a natural process as result of the tendency of materials to be in the lowest energy state. most commonly, is that the iron and steel combine with water and oxygen forming hydrated iron oxide (rust) [52].

In order to simulate the corrosion process, it is important to know the different typology of the corrosion. To our advantage, the visual classification is very important for prevention and evaluation of metallic products. The NACE that is the primary society dedicated to corrosion in the United States recognizes ten forms of corrosion [112]:

i General Corrosion: Uniformly over the surface

ii Crevice Corrosion: occurs in tight spaces between surface. Not simulated required external boundary parts knowledge, partially covered with different patterns of corrosions

iii Pack Rust: massive deformation pushing parts of the surface

iv Stress Corrosion: Rust in the cracks created by the structural tension

v Galvanic corrosion: Occurs in the boundary of two metals. Not simulated required external boundary parts knowledge, partially covered with different patterns of corrosions

vi Pitting: Perforation of the metals

vii Flash Rust: simulated with general corrosion

viii Filiform Corrosion: Build under the paint of coated metals

ix Osmotic Blistering: Form blisters in the metal coatings

x Pinpoint Rust: Splatter pattern of rust formed in the metal

Therefore, corrosion is simulated as the progressive mix of a dielectric material (corrosion) and a conductor (metal). Depending on the type of corrosion it affects in different levels (Figure 4.26):

- Distribution of dielectric material (Metalness)
- Roughness of the surface
- Surface displacement

The general procedure started collecting different images of corrosion types and create maps (texture maps) with them. The maps are *bitmap* images (a matrix whose values are between $[0, 1]$) that indicates the value of one characteristic, for instance, the roughness of the surface. These maps are unwrapped over the surface (Process described in Section 4.6.5) indicating the coordinates in the surface that correspond the map value.

We opted for this solution (image-based) instead of procedural as it allows to create easily a specific type of variation when applied to a specific industrial case, it is just to take pictures and some editing. Additionally, a procedural based algorithm will by also prompt to be identified by a powerful NN leading to over-fitting.

All the maps have to contain congruent values regarding each other and have the same *UV coordinates*. The characteristics of the maps for simulating the corrosion correspond to: (a) Diffuse color, (b) Metalness, (c) Roughness, (d) Displacement (Figure 4.26). Where the maps are the material at its maximum corroded level.

(a) The color map is an input of the color of the diffuse component in the rust material (dielectric). (b) The "metalness" map define the type of material (rust or metal) at each point of the surface, therefore its values are $1$ or $0$. (c) The roughness map contains both the rust and the metal roughness information of each point of the surface. (d) The displacement map values indicating real displacements of the mesh, the mechanism used is explained in Section 4.6.4.

The oxidation function (Figure 4.26) takes as input the maps, the level of oxidation and the base metal properties (IOR, color, and roughness). This function controls the level of corrosion that governs how strong is the appearance of the oxidation maps. This level is defined by a value called the *corrosion status*.

The *corrosion status* is an input value between $[0, 1]$ that defines the amount of corrosion of the metal. This means that $0$ is the pure metal and $1$ fully corroded, where the appearance of the material is totally defined by the maps. The blend between this two values is controlled by the oxidation function and it is explained in the next paragraphs.

The roughness of the metal is the result of the average weight (Equation 4.24) of the roughness of the map and the base metal roughness, the factor of the average is the *corrosion status*.

Similarly, the displacement is the weighted average of the map displacement and $0$ that is no displacement. The mechanism of how the displacements work is presented in Section 4.6.4.

The distribution of the rust is progressive increasing adding "particles" (small parts) of rust to the metal, given that there is not a mix of two materials. Each particle is generated using a *Perling noise* function [179] and the position of each particle is given by the *metalness* map.

The "metalness" map is a binary matrix containing the values of what parts of surfaces are rusted. We will consider that the input map $M_m$ is the most corrode state

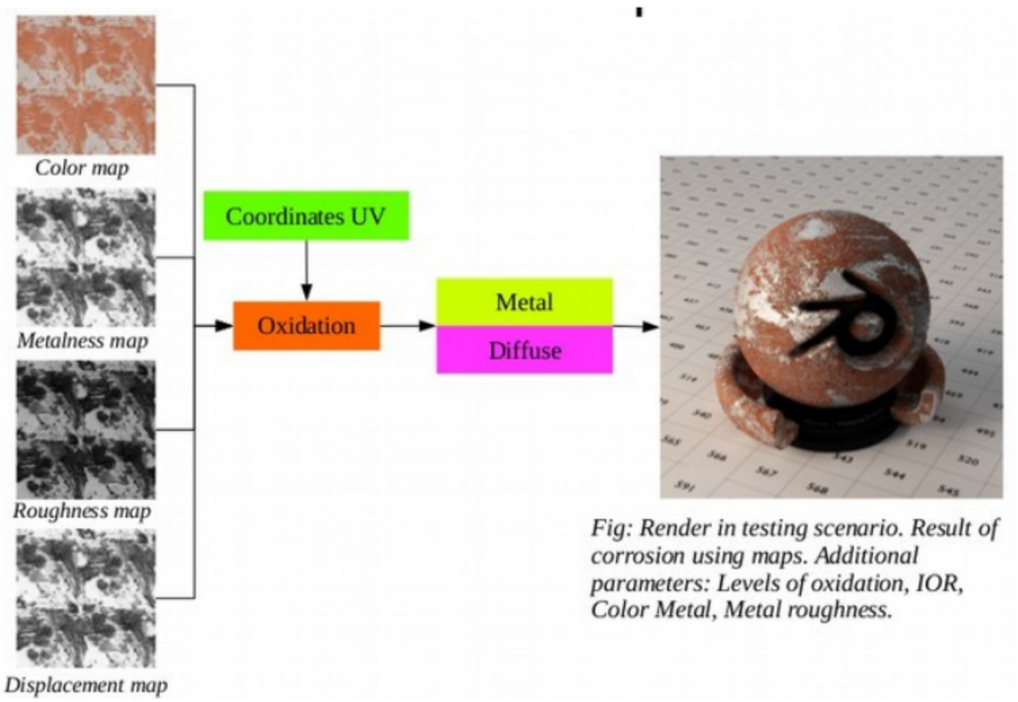

Color map
Metalness map
Roughness map
Displacement map
Coordinates UV
Oxidation
Metal
Diffuse
Fig: Render in testing scenario. Result of corrosion using maps. Additional parameters: Levels of oxidation, IOR, Color Metal, Metal roughness.

**Figure 4.26:** *General corrosion shader description. As input different maps from a type of corrosion are set into the oxidation node group that also takes an UV model coordinates. The oxidation group defines the mix between metal and dielectric*

The *Perling noise* is a *gradient noise* technique used to produce procedural textures. It is generated by assigning pseudo-random gradients to each coordinate of a dimension, then softly interpolate between the two coordinates. This technique has been used for the recreation of textures of elements of nature given its organic looking appearance.

We set a small scale value for the *Perling noise* $N_p$ generator and increase (added) to its output values a *corrosion factor*. These values are later thresholded creating a binary result, where $0$ represents the rust and $1$ the metal. Thus, the values will be $0$ or $1$ depending on the corrosion factor and the distribution of the *Perlin noise*.

The corrosion factor is between $0 \leq c_f \leq 1$, but is mapped (scaled $s$ and translated $t$) accordingly that when $c_f = 0$ there will be only metal and $c_f = 1$ totally rust. Later this noise is mixed with the initial "metalness" map generating a new "metalness" map for a given corrosion stage $M'_m$ is shown in Equation 4.26 for each pixel $(i, j)$:

$$M'_m(i,j) = \begin{cases} 1 & \text{if } (N_p(i,j) + sc_f + t) > 0.5 \text{ and } M_m(i,j) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.26}$$

The rendering result of a metal with the same type but different levels of corrosion are shown in Figure 4.28.

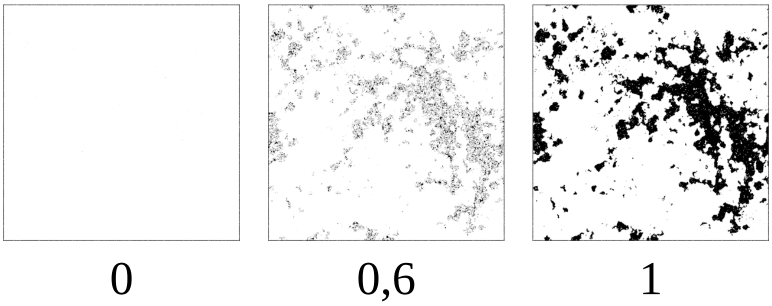


**Figure 4.27:** *Evolution of the distribution of rust (in black) over the metal (white). Controlled by the input parameter "corrosion status".*

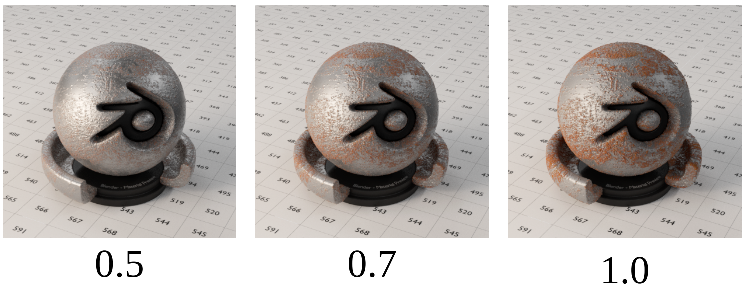


**Figure 4.28:** *Render results of metal with the same type but different levels of corrosion. Material parameters, a conductor with IOR:* $1.45$*, roughness* $0.05$*, Specular color:* $0.4$*. and Imperfection factor* $1$

### 4.6.2 Superficial imperfections

Represent the variation in the roughness of the material due to smudges, grease, dirt, minimal scratches. The process is similar to the one followed in the corrosion where a map (roughness map $M_r$) with values between $[0, 1]$ defines the value of the roughness. This map is multiplied by a value $i_f$ (Imperfection factor) that indicates the strength of the superficial imperfection, such that $0 \leq i_f \leq 1$). When the factor is $1$ the map affect totally the roughness of the object:

$$M_r' = i_f M_r \tag{4.27}$$

In order to consider the previous roughness values $M_p$ (the mix between the base material and the corrosion roughness), the new roughness map $M_r'$ will be the factor that defines the mix between the previous roughness $M_p$ and the initial roughness map $M_r$. In this way when the imperfections factor $i_f = 0$ the roughness of the material is defined by the base or rust roughness $M_p$. And when $i_f > 0$ the roughness is averagely weighted of the previous roughness $M_p$ and the roughness map $M_r$ using as weights the new roughness

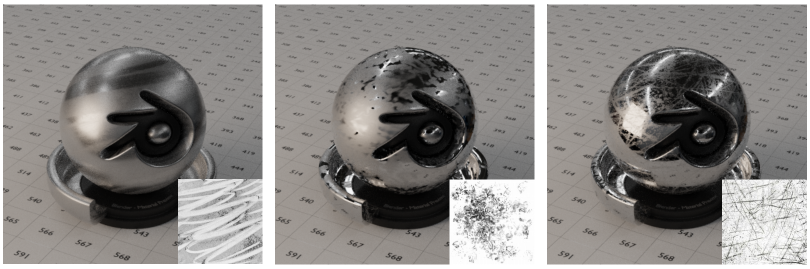


**Figure 4.29:** *Superficial imperfections generated by three different maps (left corners). Material parameters, a conductor with IOR:* $1.45$, *roughness* $0.05$, *Specular color:* $0.4$. *and Imperfection factor* $1$

map $M_r'$.

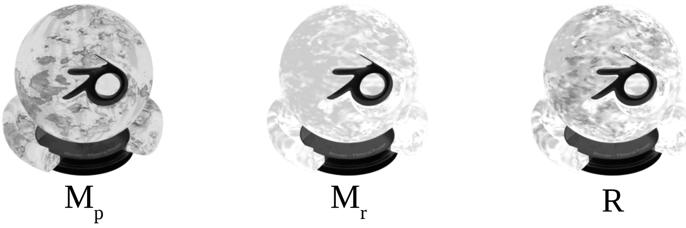


**Figure 4.30:** *Roughness map mix* $R$ *between previous roughness map* $M_p$ *and imperfections roughness map* $M_r$

In this way both roughness are take into account. By replacing in Equation 4.24 the calculated roughness $R$ is computed with the equation 4.28. In this case all operation are element wise operations (Figure 4.30).

$$R = M_r \circ M_r' + M_p|M_r' - 1| \tag{4.28}$$

An example of the produced renders of three different roughness maps is shown in Figure 4.29. All have the same factor $i_f = 1$. The material used was basic conductor with IOR: $1.45$, roughness $0.05$, Specular color: $0.4$.

### 4.6.3 Grime

The grime is a superficial layer of other material added to the main one. The material of the grime is simulated as dielectric, some examples are grease or dust. The configuration is similar as for how the corrosion works. Where there is one material (Grime) that is overlaid over the main material distributed according to a map (Figure 4.31).

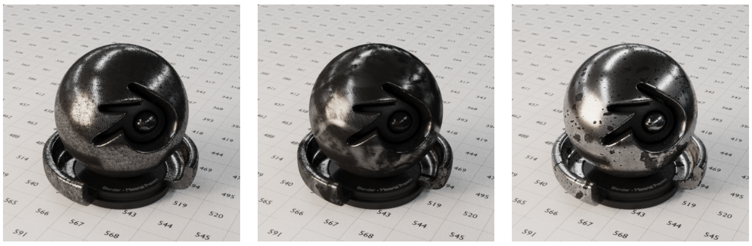


**Figure 4.31:** *Example of renders produced with different grime maps. The material used was basic conductor with IOR:* $1.45$*, roughness* $0.2$*, Specular color:* $0.4$

The progression of the grime is defined by the parameter $0 \leq g_f \leq 1$ (Grime factor). And the thresholded grime map $M'_g$ that govern the distribution of the grime is given by the equation 4.29. Where the initial grime map $M_g$ is weighted by the grime factor and thresholded:

$$M'_g = \begin{cases} 1 & \text{if } (g_f M_g + |g_f - 1|) > 0.6 \\ 0 & \text{otherwise} \end{cases} \tag{4.29}$$

The resulting map $M'_g$ defines the distribution of the grime over the main material. For each pixel of $M'_g$ if the value is $0$ the material assigned to the object is the grime otherwise the material is the main. Notice that the main material could be the one with already other variations such as oxidation.

### 4.6.4 Micro-displacements

The micro displacements are real displacements of the model surface (mesh) and the main difference with the surface imperfections is that the micro displacements are meant to simulate deformations of the mesh (up to 4mm). The displacements are performed at rendering time where the vertex of a mesh is translated along its normal (the vertex normal [79]).

The amount of the displacement is proportional to a map (Displacement map) this map contains values between $[0, 1]$, one defining the parts where there are a maximum displacement and $0$ where there are not displacement.

This map is multiplied by a constant that defines the amount of displacement (Displacement factor $d_f$) and a constant that scales all the values defining the maximum displacement. Additionally, the displacements are added to the displacements coming from the oxidation process.

In order to obtain high-resolution meshes, a *Subdivision Surface* operation is applied to the models before rendering. A *Subdivision Surface* is a recursive operation in which the mesh faces are divided into smallest ones in order to create a smooth surface or adding resolution to the mesh. We applied a simple subdivision in which only more vertex is added to the mesh without altering the model appearance (Simple subdivision).

Besides for improving the performance and use of resources, a *Adaptive Subdivision* were used. This technique subdivides the meshes based on how far they are from the camera. Thus the polygons of the surface are divided into smaller ones (*micropolygons*) whose size is controlled by their screen projection and a dicing rate [43].

The final result of micro-displacements is shown in Figure 4.32. Where a basic metal is subject to different deformation patterns.

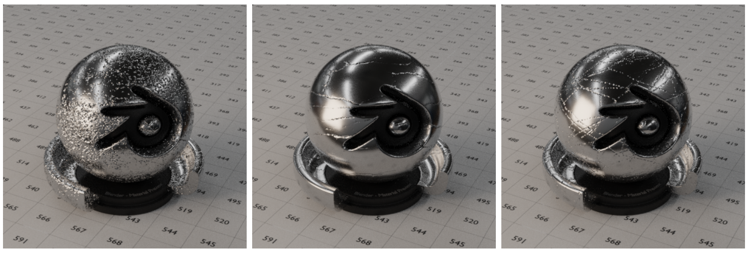


**Figure 4.32:** *Example of renders produced with different displacement maps. The material used was basic conductor with IOR:* $1.45$*, roughness* $0.2$*, Specular color:* $0.4$

### 4.6.5 Texture Mapping

A commonly used method for adding visual details to a 3D model is to map patterns onto their surfaces. These texture patterns could be defined by *procedural textures* or by a matrix of values (bitmaps). The methods for assigning this maps to the surfaces are referred as *texture mapping*. Where a function $\Phi$ maps from the surface $S$ to the texture space $T$ (Equation 4.30) [151].

$$\Phi : S \to T \tag{4.30}$$

The texture space $T$ is a rectangle where is placed the texture map and usually is normalized and represented by the coordinates $(u, v) \in [0, 1]^2$. The texture coordinate function $\Phi$ defines how to warp the texture map to fit the geometry of an object surface. There are two main approaches for defining this function, geometrically for simple cases (planar, spherical, cylindrical or cubic projections), and interpolation of texture coordinates [151].

In the interpolation of texture coordinates, each vertex of a triangle mesh is projected onto the texture space and the values inside the triangle are calculated using barycentric coordinates. The quality of the "unwrapping" is given by where are assigned the vertex in the map and the formal coherence regarding the simulated object.

This allows to assign values from a *texture map* to each point of the surface mesh and having a finer control (by *texel*) than by each vertex.

In our case, all the maps that belong to the same extrinsic variation share the same UV coordinates. The process of assigning the UV coordinates (*UV mapping*) usually is performed manually by skilled designers. Nevertheless, we propose an automatic process to generate synthetic datasets given that for each class hundreds of 3D models may be required to be processed.

Therefore, we assign automatically the coordinates using the Blender's "Smart UV Project". This method analyzes the input mesh and cuts it based on the angular changes of the faces. It is suggested for complex geometric forms such as mechanical objects or architecture [29].

In order to create more variations and avoid over-fitting to the texture of the models, the UV coordinates of each variation are randomly modified, by applying a rotation and scaling transformation in each sample generation.

Finally, all these variations are integrated together in two main materials (dielectric and metal with variations). This configuration exposes the map corresponding to each variation and its respective factors. The map and the factor are randomly drawn from the distribution specified in the configuration file in order to create a dataset with different statistics.

## 4.7 Environmental Variables

The environmental variables are composed by three main elements: a *direct lighting* that is the main source of illumination of the object, the *environment light* that simulates the indirect lighting coming from the surroundings and the *external elements* which are other objects different to the ones to be recognized.

### 4.7.1 Lighting

Two different sources of light are used in the scene, the main (direct) and environment (indirect) lighting. The **main light** is simulated as a plane the emit light at the normal direction.

The variables that control the main light will randomly drawn from a specified distribution from the configuration file and are defined as:

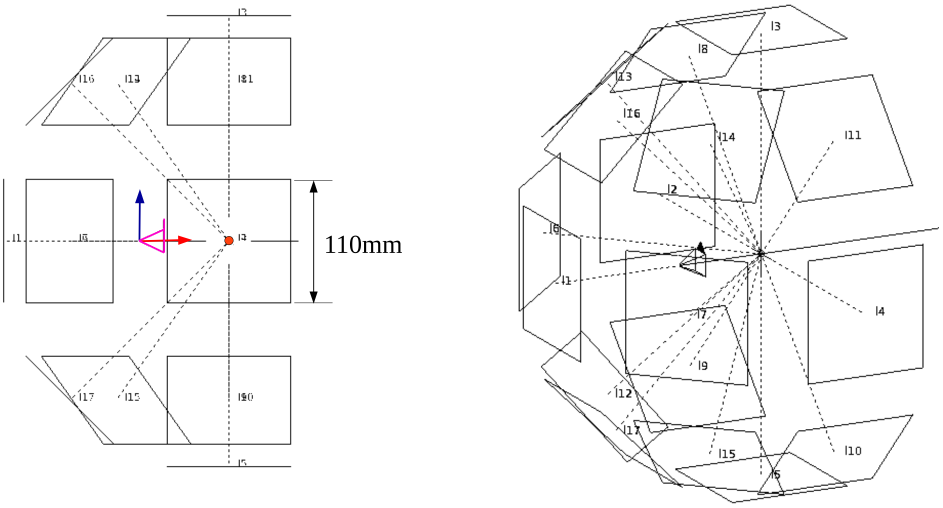
**Direction** is defined as the angle formed by the camera point direction and the normal of the light. Hence, is a global direction as the camera remain static in all the samples and the direction of the light will be regarding to the observer. The directions will be the same no matter the rotation of the objects, but will always be pointing at the center of the main object (Figure 4.33).

Additionally, only one of the lights will be turned on for render the rest of the lights will appear as "invisible" in the rendering process.

**Intensity** defined as the radiant intensity of the plane (irradiance). Thus, we use the visual flux density at $555\mathrm{nm}$ the peak of the spectral sensitivity of the human eye [153]. The intensity will be expressed in lumens using the conversion factor ($1000\mathrm{W/m}^2 = 683\mathrm{lm/m}^2$ at $555\mathrm{nm}$).

**Color temperature** The tint of the light is defined by the temperature($\mathrm{K}$) of a blackbody radiation model (Section 4.3.3). Incandescent lights resemble the theoretical blackbody as they emit a continuous spectrum of all visible colors of light, therefore, the color temperature accurately describe the incandescent spectrum [81].

In the scene, 17 main lights are used and only one is used randomly chosen at the rendering time in a specific POV. The lights are distributed in the hemisphere around a point called the "light center". Thus all the lights normals are pointing the lights center. The center of the object will always be equal to the center of the main object (Figure 4.33).



(a)

(b)

**Figure 4.33:** *Blender main scene. (a) Front view of the main scene. On red global $x$ axis, blue global $z$ axis, on purple the camera and on orange the center of the lights. The dashed lines indicate the normal of the light planes. (b) Orthographic view of the main scene. In the scene,* 17 *main lights are used and only one is used randomly chosen at the rendering time in a specific POV. The lights are distributed in the hemisphere around a point called the "light center". Thus all the lights normals are pointing the lights center. The center of the object will always be equal to the center of the main object.*

The **Environment lighting** is used to simulate light coming from every direction (direct and indirect) as if the object were placed in some scene without the need to recreate all the elements. We use the approximation (Image-Based Lighting (IBL)), that use light proves images that records omnidirectional in HDR images of the incident illumination conditions in a particular point in space (Section 4.3.8).

The simulation of indirect lighting is important because metals do not present diffuse shader which means that in soft surfaces (low roughness) the incoming and outgoing direction angle regarding the surface normal tend to be equal. Thus most of their appearance is given by the mirror-like reflection of their surroundings. An example of renders produced with this configuration is shown in Figure 4.34

### 4.7.2 Occlusion and External Elements

In order to recreate realistic occlusions and cluttered scenes that are congruent in terms of all the parts have the same lighting, reflections, and shadows. Different external elements (Elements not included in the dataset) were added to the scene together with the main object.

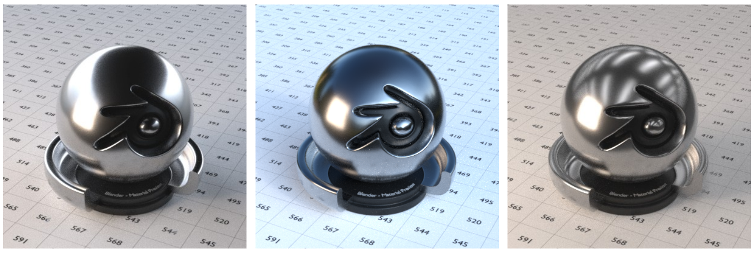


**Figure 4.34:** *Example of renders produced with different environment light probes. The material used was basic conductor with IOR:* $1.45$*, roughness* $0.2$*, Specular color:* $0.4$

In contrast with other methods for generating occlusion where parts of the main elements are covered or removed [254]. The proposed method creates distracting elements that share the same environmental conditions in order to simulate realistic scenarios.

The external parts material are treated equally as one of the main objects, where their parts have different materials that could be metal or dielectric with its respective variations (intrinsic and extrinsic). The only difference is that their parameters are picked with a random distribution.

The number of external elements to load for each sample is defined previously in the configuration file with a type of distribution that indicates the possibility of each number of elements.

Further, the scale of the external elements is performed in such a way that the larger dimension of the object fit a specified dimension (limit size). A limit size is a number randomly picked between two limits previously specified.

The process of position the external parts in the scene is performed in two subprocesses, (i) position the part inside the visible volume, (ii) check collision with other parts in the scene.

For the first subprocess, the visible volume is the area of the scene inside the visible limits of the camera (Illustrated in Figure 4.19). First is computed the position of the object $x_1$ in the axis of the direction of the camera $x$. The random position in this axis is limited by two limits $[L_1, L_2]$ that define how close and how far the object is from the viewer.

After the position in the axis, $x$ is defined, the position in the other axis is defined in the same way. A random value is computed inside two limits, but in this case, the limits are set by the visible limit at the $x$ coordinate. As the rays of the camera can be traced with a linear equation, the limits of the other axis are $[-mx_1, mx_1]$ as shown in Figure 4.35.

The slope of the camera $m$ is obtained by dividing the half of the sensor width $W_s$ with the focal length $f$ (Equation 4.31)

$$m = \frac{W_s/2}{f} \tag{4.31}$$

For the second subprocess, a routine tries to load the objects and avoid collisions with

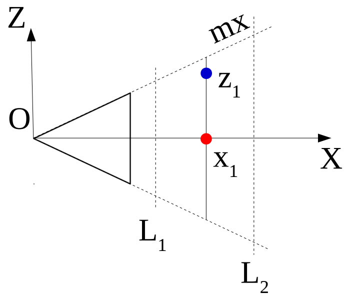


**Figure 4.35:** *Random coordinates $(x_1, z_1)$ for position external elements inside visible volume (in gray). First the coordinate $x_1$ is computed inside the limits $[L_1, L_2]$. The position $z_1$ is computed inside the limits $[-mx_1, mx_1]$.*

the already objects present in the virtual scene. A Bounding Box (BB) collision check, run each time a part is randomly positioned if it is in a collision, it positions the part again in another random position. It repeats this operation for a defined number of times. In the end, there is the possibility that the part is in a collision but is saved in the label of the render.

The BB collision check compare the BB representation of two objects. Each one of the vertexes of the is checks if are inside the volume of the other BB. Thus a new object to be placed is compared with each object present in the virtual scene.

Finally, after the render is finished the pixel-wise segmentation, that indicates which pixel belongs to the main object is saved (Figure 4.21b). This file is a matrix of the same dimensions of the rendered image (Figure 4.36). It is acquired by setting an index to the main object that is later returned when a camera ray hits the object at rendering time.

### 4.7.3 Automatic model preparing

The model fixer is a semi-automatic routine that is aimed to prepare a 3D model for rendering. It is semi-automatic because it displays the model and asks for a user to tell if the final result is good enough, and if not it tries different patterns of importing (Figure 4.37).

This routine is composed of the next steps:

i Import file: input files used are in DAE format

ii Clear parts parent relationships. The parent relationships constrain the rotation, position, and scale of an object to the transformations performed to another object (the parent). When the object is imported, the parent relationship between their parts is cleared. Two types of clearing can be performed, conserving the transformation of the child objects or that they return to the original position/scale/rotation.

First is performed one and asked the user if the object parts are located correctly if not it tries the other parent cleaning.

iii Join all the parts into a single mesh

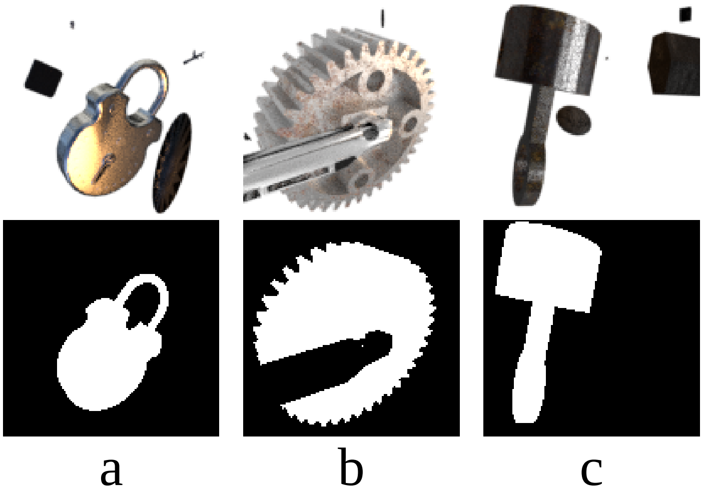


**Figure 4.36:** *Example of external elements and respective pixel wise segmentation. Main objects (a) lock, (b) bearing, (c) gear.*

iv Remove doubles vertex: merge the vertex that is at a distance less than a defined value.

v Convert triangles to quads model: convert all the triangles of a mesh to quadrangles by removing the shared edge between two adjacent triangles.

vi Make normals consistent: make all the face normals to face outside the object

vii UV mapping: assign UV coordinates to the mesh, the method used is the Blender's "Smart UV Project". This method analyzes the input mesh and cuts it based on the angular changes of the faces. It is suggested for complex geometric forms such as mechanical objects or architecture [29].

viii Remove external elements except for the main object, all the parts that were not joined in the main mesh are removed

ix Rename mesh equal to save filename

x Assign the model center to be the geometry centroid

xi Apply all transformations (Location – Rotation - Scale) made to the object

xii *Smooth Shading*: surface normals will be interpolated at shading time, thus surfaces will appear smooth

xiii *Edge Split*: modify the vertex normals according to the edge angles to make the edges appear sharp

xiv *Surface Subdivision* to add resolution to the mesh

xv Save Blend file with name equal to mesh

The materials are imported as assigned for the designers in the original files. Later when generating the renders, each material is overridden for the ones presented here. This allows keeping the link between the mesh and materials division.

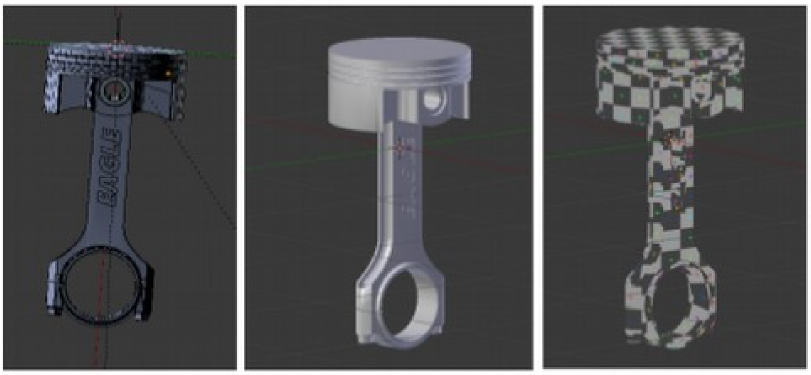


**Figure 4.37:** *From left to right: (i) Imported .dae file from 3dwarehouse.sketchup.com.(ii) Prepared model (iii) Prepared model displaying UV coordinates.*

### 4.7.4 Data Augmentation

*Data Augmentation* is a commonly used technique for increasing the number of variations of datasets without the need acquiring new data. This is achieved by applying a set of transformations to each one of the samples of a dataset.

In our case, the dataset is produced without any data augmentation but instead, it is performed in real time during training allowing that each sample that is used for training to be different. Thus it is avoided to save unnecessary data and be flexible to different types of augmentation depending on the task.

For instance, a commonly used transformation is to rotate each sample at a random angle. If this is performed the information of position and rotation of the camera regarding the part is corrupted and can not be used for training a model for *registratrion*.

Therefore in order to allow the data augmentation in training time, all the samples are saved with the alpha channel to allow the addition of backgrounds, a segmentation mask and all the information used for generating each sample.

## 4.8 Discussion and Future Work

In this chapter has been presented a method for the creation of fully labeled synthetic datasets for training and evaluating surrounding understanding algorithms. It is derived from the physically based recreation of proposed relevant characteristics classified as intrinsic, extrinsic and environmental that typically influence surrounding understanding methods.

Therefore it is based on typical factors that influence the visual characteristics of the objects found in the literature. Specific cases in the industry could happen where there are other factors, but the same method applies in order to recreate the domain variable of interest.

Each one of the parameters that define the characteristics of the renders is drawn from determined Probability Density Function. This configuration allows to define ranges of possible values that could take some properties of the real-life in physically accurate terms, and how often is expected to find these values in a specific case.

Additionally, this defines the bias of the datasets in terms of high level of characteristics, such as level of corrosion of the parts of the typical view of the objects. Instead of low-level characteristics such as the histogram of colors or pixel by pixel comparisons.

Regarding the level of accuracy of the physical models used for the recreation of the reality, there will be always a balance between deep and agility in terms of implementation and performance. In our case, we have opted for simulating the most relevant factors (according to literature) that cause larger and broader impact. For instance, smaller physical phenomena (e.g. iridescence or caustics) were not considered as there were not favorable in cost-benefice terms of visual impact.

Another point to consider is that the input assets (maps, 3D models and light probes) define the bias and well performance of the proposed method. In the case of the texture maps, we opted for this solution (image-based) instead of procedural as it allows to create easily a specific type of variation when applied to a specific industrial case, it is just to take pictures and some editing. Additionally, a procedural based algorithm will be also prompt to be identified by a powerful NN leading to over-fitting. And the same applies to the light probes.

Nevertheless the main issue will arrive when it is not aimed to recognize a particular case but instead a broader domain, for instance, all the nuts from *ImageNet* dataset. The number of the assets and how well they fit the target domain statistics will play an important role.

Further, automatic *UV mapping* play an important role given that it defines the location of variations of the objects. The current implementation is purely based on geometric characteristics and it is the same for all the variations. A future development could be based on the implementation on a smart location of variation maps depending on the type of the variations and distribution around the parts.

In many cases, the background where is placed an object could reveal information to the surrounding understanding algorithm about the object. However, this information is weak and could lead to learning features that are not by itself of the objects. This leads to one of the major issues of *feature learning* techniques, that no one is really sure about the features learned by the models. Therefore they could present a good performance in laboratory tests, but in real-life miss-labeling could arise from weak features in novel scenarios.

On the other hand, simulating coherent scenes will allow algorithms to learn the interactions of the environment and objects present in the scene. Still, we believe that this could be better achieved by a separate learning process. In the first one, robust characteristics of the objects are learned. In the second one, interpretation of the surroundings and the role of the objects could be inferred. Therefore, the presented method is located in the first type of learning where only is simulated light-matter interactions.

Hence, the synthetic datasets could be an indirect form of controlling the features learned by the NN based models by controlling the type and characteristics of the input data.

Finally, this represents a method for easily generate training data that otherwise will be tedious such as pixel-wise segmentation or Point Of View (POV) that takes into account realistic variations of the objects. In many cases, it could be mixed with other methods, for instance, enhancing a real photos dataset or other DA methods.

In the next Chapter will be presented a study case of the proposed method for the task of object classification framed into a domain adaptation problem.

## 4.9 Appendix

### 4.9.1 Config file and DV distribution

Each one of the values of the DV such as light intensity, POV or material type among others, at render time is drawn from a PDF specified in the configuration file. An example of configuration file (Exponential distributions).

```
{
  "saveFolder" : "/tmp/renders/",
  "modelFolder" : "/3D_models/models/",
  "ext_model_folder" : "/3D_models/ext_models/",
  "envFolder" : "/Environment/",
  "materialsFolder" : "/Material/",
  "texturesFolder" : "/Material/textures/",
  "main_blend_file" : "/Cycles/Scene.blend",
  "classes_to_render" : ["nut", "shock", "key"],
  "samples_render" : 10,
  "img_res" : 128,
  "render_processor" : "GPU",
  "render_tile_size" : 128,
  "num_pov" : 1000,
  "num_rand_var" : 1,
  "starting_pov" : 0,
  "metal_probability": 0.5,

  "pov_pos_distribution" : "normal",
  "pov_rot_distribution" : "normal",
  "pov_pos_sigma" : 100,
  "pov_pos_mean" : [800,0,0],
  "pov_rot_sigma" : 40,
  "pov_rot_mean": [0,0,0],

  "light_distribution" : "random_simple",
  "light_int_lim" : [2600, 3300, 100],
  "light_temp_lim" : [3000, 5500, 500],
  "environ_int_distribution" : "normal",
  "environ_int_mean": 2.5,
  "environ_int_sigma": 0.2,

  "num_ext_elem_distribution": "random_simple",
  "ext_elem_lim" : [0, 5, 1],

  "color_distribution": "random_simple",
  "color_dielectric_lim" : [0.1, 0.9, 0.1],
  "color_metal_lim" : [0.5, 1, 0.1],

  "ior_distribution": "random_simple",
  "ior_dielectric_lim" : [1.3, 1.7, 0.1],
  "ior_metal_lim" : [0.5, 4, 0.1],

  "corrosion_distribution": "exponential",
  "corrosion_beta": 0.1,

  "grime_distribution": "exponential",
  "grime_beta": 0.1,

  "gloss_distribution": "exponential",
  "gloss_beta": 0.2,

  "displacement_distribution": "exponential",
  "displacement_beta": 0.2,

  "roughness_distribution": "normal",
  "roughness_mean": 0.1,
  "roughness_sigma": 0.05
}
```

### 4.9.2 Renders Example

Examples of the renders produced with the presented method of recreating domain variables (Figure 4.38).

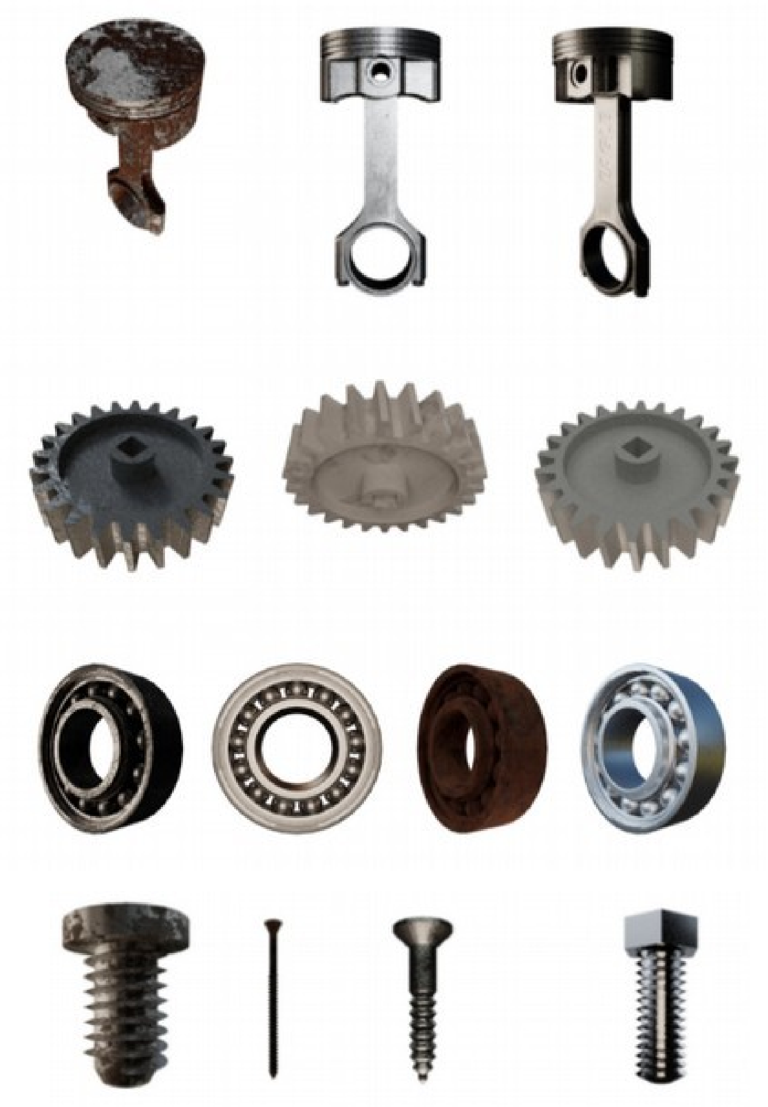


**Figure 4.38:** *Example of renders produced with the presented method.*

CHAPTER 5

# Convolutional Neural Network for Domain Adaptation a Study Case

## 5.1 Introduction

In this chapter is presented a series of experiments using the proposed method for the creation of synthetic datasets for training and evaluating *surrounding understanding* methods. Thus the proposed synthetic dataset could be used for other tasks such as *segmentation* or *registration*.

Nevertheless, this study will be focused on object detection due to the importance of Industrial Augmented Reality (IAR) which is the frame of this work. Where CNN models are trained with synthetic data in order to recognize real-life objects.

The differences with most of the previous research in the field of Domain Adaptation (DA) is that usually is available a small representation of the target domain (real photos) [45]. In contrast, this research deals with training solely with synthetic datasets, but specifically, it is aimed to determine the influence of the variation of *domain variables*

Previous research has stated that increasing the realism in the synthetic datasets increase the performance [160,178,251] and that mixing synthetic data with the real images surpass the performance of the trained models with only real data. Even if the synthetic data is in a lowest level of realism (*wireframe*) [160].

Similarly Physically Based Shading (PBS) have been proposed [116,251] showing that it improves the performance over other *shading* methods. Nonetheless, these research use perfect materials, in optimal conditions.

Another approach could be to create the data using random variations of some of the characteristics of the domain without the intention of creating realist results (*Domain randomization* ). By selecting a property and randomly assign values creating high levels of variation in the dataset. This approach proves that using large enough datasets the performance achieved is useful for some visual tasks [214].

Although it is possible to create samples by assigning totally random values to each part of the surface of the objects, the probability of creating patterns useful patterns will be immense. Thus for *domain randomization* it is necessary to define rules for the variations. [214] proposed to variate procedural textures (checkboard or gradients). It is hypothesized by the authors that realistic variations could lead to improving the results because the patterns recreated are similar to the ones found in real photos.

Following that line of ideas and that one of the keys to obtaining realistic renderings is the simulation of the imperfections of reality, in Chapter 4 is proposed a method for the creation of synthetic datasets that recreates relevant domain variations using a PBS approach accounting the distribution of the variations.

From a *Domain randomization* perspective this approach is based on setting real-life based rules in the possible variations of the domain. In order to recreate feasible patterns that occur in reality, shorten the distance between domains.

Additionally, this applies to objects were their main visual variations are due to the physical-chemical process of the rough materials. Another type of variations found in products could be due to artistic textures in objects. But in general, the proposed method could be applied to another type of elements were the main factors of the visual variation could be identified.

### 5.1.1 Domain Adaptation (DA)

The problem faced in this research is framed under the field of *Transfer Learning (TL)*. In this section is presented a formal definition and notation of [45, 174, 239].

A *Domain* $\mathcal{D}$ is described by two parts, a feature space $\mathcal{X}$ and a marginal probability distribution $P(\mathbf{X})$ where $\mathbf{X} = \{x_1, ..., x_n\}$. The feature space $\mathcal{X}$ is composed of all possible features and $\mathbf{X}$ is a particular learning sample set of the domain with $n$ number of features.

For a given domain $D = \{\mathcal{X}, P(\mathbf{X})\}$, a *task* $\mathcal{T}$ could be defined by two components as well, a label space $\mathcal{Y}$ and a predictive function $f(\cdot)$ trained from the feature-label $x_i, y_i$ where $x_i \in \mathbf{X}$ and $y_i \in \mathbf{Y}$. Where $\mathbf{Y} = \{y_1, ..., y_n\}$ are the corresponding labels of the particular learning sample set $\mathbf{X}$.

The function $f(\cdot)$ predicts the corresponding label $f(x)$ of a new feature $x$. The predictive function can be seen as $P(y|x)$, the probability of $y$ given an unseen feature $x$.

In a general case, we have two domains with their related tasks, the *source domain* $\mathcal{D}^{\mathcal{S}} = \{\mathcal{X}^{\mathcal{S}}, P(\mathbf{X^S})\}$ and its respective task $\mathcal{T}^{\mathcal{S}} = \{\mathcal{Y}^{\mathcal{S}}, P(\mathbf{Y^S}|\mathbf{X^S})\}$. Similarly, the *target domain* $\mathcal{D}^{\mathcal{T}} = \{\mathcal{X}^{\mathcal{T}}, P(\mathbf{X^T})\}$ with $\mathcal{T}^{\mathcal{T}} = \{\mathcal{Y}^{\mathcal{T}}, P(\mathbf{Y^T}|\mathbf{X^T})\}$.

Therefore Transfer Learning is defined as the process of improve the target predictive function $P(\mathbf{Y^T}|\mathbf{X^T})$ using information from $\mathcal{D}^{\mathcal{S}}$ and $\mathcal{T}^{\mathcal{S}}$ with the condition that $\mathcal{D}^{\mathcal{S}} \neq \mathcal{D}^{\mathcal{T}}$ or $\mathcal{T}^{\mathcal{S}} \neq \mathcal{T}^{\mathcal{T}}$. When both (source and target) are equal, the problem correspond to a classical ML application.

In DA methods, the task in the *source* and *target* are the same $\mathcal{T}^{\mathcal{S}} = \mathcal{T}^{\mathcal{T}}$ (transductive TL). However the data representation is different or have different distributions $\mathcal{X}^{\mathcal{S}} \neq \mathcal{X}^{\mathcal{T}}$. According to this definition label sets are shared $\mathcal{Y}^{\mathcal{S}} = \mathcal{Y}^{\mathcal{T}} = \mathcal{Y}$ and the predictive functions are the same $P(\mathbf{Y^T}|\mathbf{X^T}) = P(\mathbf{Y^S}|\mathbf{X^S})$. In real application the second supposition rarely holds, therefore the DA covers the cases where only the labels are shared.

### 5.1.2 Hypotheses

We will refer to two domains, the real (target $\mathcal{D}^{\mathcal{T}}$) and synthetic (source $\mathcal{D}^{\mathcal{S}}$). The target domain is defined by a dataset that contains images obtained from photos of the objects. The source refers to the one created with the proposed method (Chapter 4). Each one is referred as a different domain ($\mathcal{D}^{\mathcal{S}} \neq \mathcal{D}^{\mathcal{T}}$) with its own marginal probability distribution $P(\mathbf{X})$) and their share the same task $\mathcal{T}^{\mathcal{S}} = \mathcal{T}^{\mathcal{T}}$.

i The physically-based recreation of the possible variations of objects in real-life will recreate also the visual patterns generated by the light-matter interaction that are learned by NN models. This will lead to obtaining better performance results than training in their absence (in datasets with similar characteristics).

ii If two domains (source and target) are closely related, the differentiability of the objects will be independent of the type of training (with synthetic or real images). Performance no matter the type of training will depend on the chosen objects. Therefore, the same groups of classes will have the same type of behavior in both domains.

iii Higher resolution samples contain more detailed information, therefore the difference between performance between domains will be proportional to the resolution of the samples.

iv Pre-trained models in large real-life datasets learned basic features (edges, corners, and curves among others) in their convolutional layers. Fine-tuning this model will boost the performance but the domain performance difference will continue as this differences should be caused by the characteristics difference but not for the recreation of the basic features.

v In domains highly related, the updates of the parameters in the learning process of NN using one domain, should affect the performance of both domains in equal directions.

vi The major difference in performance of NN trained in different domains are due to the mismatch of domain characteristics if they are coherent regarding the laws of physics that govern the samples.

In the next Section 5.2 will be presented the architectures of trained CNN models, Next the experiments design in Section 5.3 followed by the creation of the synthetic dataset (Section 5.4) and finally the results are presented in Section 5.5

## 5.2 CNN Architecture

The used architecture for the tests is MobileNet [106]. MobileNet is an efficient CNN architecture proposed to be used mobile vision applications, thus it fits the needs of the framework of this thesis in Industrial Augmented Reality (IAR) applications. In this section a brief introduction to the architecture is presented.

The core MobileNet architecture is their proposed *depthwise* separable filters. Where the main intention of is to reduce the required computational power of traditional convolution filters.

The depthwise separable filters are composed by two standard convolutions, (i) Depthwise convolution, which is applied to each one of the channels of input and (ii) Pointwise convolution, that is a 1x1 filter used to combine the outputs of the depthwise convolution.

In classic CNN architectures the convolution filter $K$ perform both filtering and combining in a single step. Where the filtering process applies a set ($N$) of convolutional kernels (dimensions $(D_K, D_K, M)$) and combine when stacking the results of the convolution of an input feature map $F$. Therefore, the next filters are required to have an equal depth of the previous activations (dimensions $(D_F, D_F, M)$) that the produced feature map $G$ (Figure 5.1).

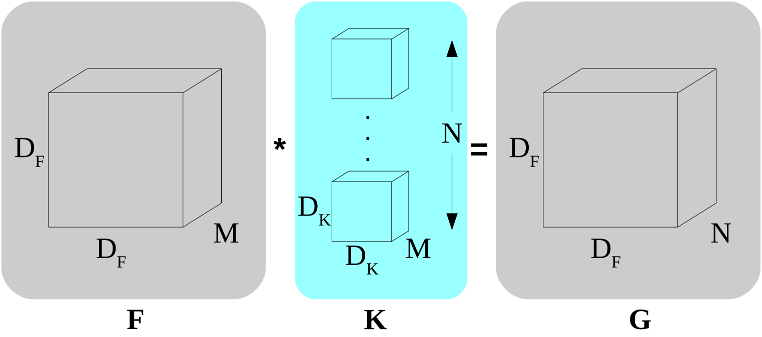


**Figure 5.1:** *Classic convolution operation in CNN. An input feature map $F$ with dimension $(D_F, D_F)$ and $M$ channels is convoluted by the set of filters $K$ producing a feature map $G$ with dimensions $(D_F, D_F, N)$*

On contra proposition in MobileNets, each operation of the convolution (filtering and combining) is performed in a different layer. First, a convolution for each channel is performed independently (depthwise $\hat{K}$), therefore the number of kernels is equal to the number of input channels. Later the activation of these kernels is combined using pointwise convolutions $\hat{G}$ with $N$ number of filters (Figure 5.2).

Therefore, this architecture relies all on its computations in 1x1 computation that is highly optimized. Additionally, each one of the convolutions is followed by a batch normalization [111] and a RELU nonlinearity.

The total architecture is formed by initially a regular convolution, followed by depthwise filters and a final fully connected layer to sum 28 layers. In our case we used *Adam* optimizer [123] and the width and resolution multipliers as default $\alpha = 1, \rho = 1$.

Finally, in our experiments, we used a fine-tuning technique, where all the initial weights of the convolutional filters are set from a pre-trained model and are not updated in the training process. And only the last fully connected layer is updated in the training process with the images from the new domain.

## 5.3 Experiments

The basic configuration of all the experiments consist of training several CNN models with different source datasets under the same conditions (classes, resolution, initializa-

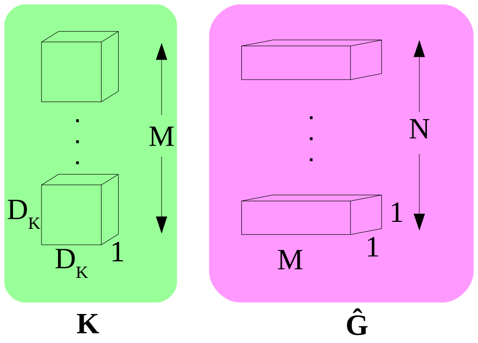


**Figure 5.2:** *MobileNet depthwise convolutions. For each channel is performed independently (depthwise $\hat{K}$), with the number of kernels is equal to the number of input channels $M$. Later the activation of these kernels is combined using pointwise convolutions $\hat{G}$ with $N$ number of filters.*

tion). Then evaluate the predictions of the models taking as input data the target dataset (Figure 5.3).

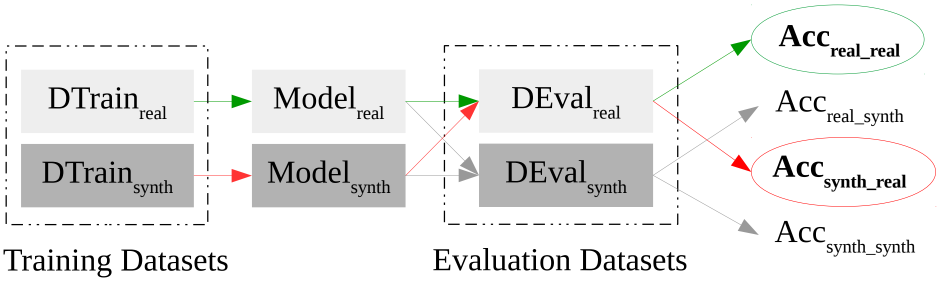


**Figure 5.3:** *Basic configuration of experiments. Two models ($Model_{real}$ and $Model_{synth}$) trained with its respective datasets and cross evaluated the evaluation datasets. The results are the Acc that stands for accuracy. In red and green are process of mayor interest, compare the evaluation of models with training with synthetic vs real.*

The training of this two models (Figure 5.3) are performed using the same conditions:

- Classes: Number and type
- Samples Resolution and all the images were square (weight=width)
- Number of times each sample feed the training of the model (Epocs).
- Random initialization, the random initial weights are the same for both models
- Fine-tuning: initial weights are taken from a pre-trained model.
- Architecture type: internal configuration of the models

– Regularization

This basic configuration is performed several times picking random classes and training both models. At the same time, fixing other parameters. Therefore the next variables were considered across the experiments:

i Model of the same class

ii Number of models

iii Training samples resolution

iv Epochs

v Fine-tuning

vi Extrinsic variables distribution (Section 4.6)

vii Intrinsic variables distribution (Section 4.5)

viii Environmental variables

The experiments were divided into two main groups, the first one (**Unknown characteristics**) without knowing the target domain characteristics or any information about the classes, and the second with a crafted dataset (**Known characteristics**) with *known characteristics* of their parts such as materials, backgrounds and extrinsic variations.

Further, the difference between the two types of experiments is that in the first one (*Unknown characteristics*) the marginal probability distribution is totally unknown, while in the *known characteristics*, the marginal probability distribution is partially known as some of its characteristics and the possibility that they appear is known.

Additionally three datasets sources will be compared:

- Synth: the proposed method that in the case of the experiments with the *unknown statiscics* is generated using three difference configuration aimed to produce datasets with different marginal distributions
- Clay: is a state of the art, non-realistic rendering technique. It sets a dull gray material for the objects and only diffuse reflections are calculated.
- Photos: is a subset (70%) of the target domain dataset. In the experiments of the *unknown statiscics* is referred as *imgnet*.

Therefore, the datasets generated with the proposed method (synth) will have the same feature space $\mathcal{X}$ but different marginal probability distribution $P(\mathbf{X})$. They are aimed to present a more similar feature space than the Clay rendering with respect the target domain. On the other side the photos will have the same feature space and marginal distribution that the target.

### 5.3.1 Known Characteristics

The experiments of the *known characteristics* are aimed to recreate an "industrial scenario", where there is supposed to have the requirement of identifying the parts of a *crosshead gimbal*. This object is composed of six parts, four metal components and two

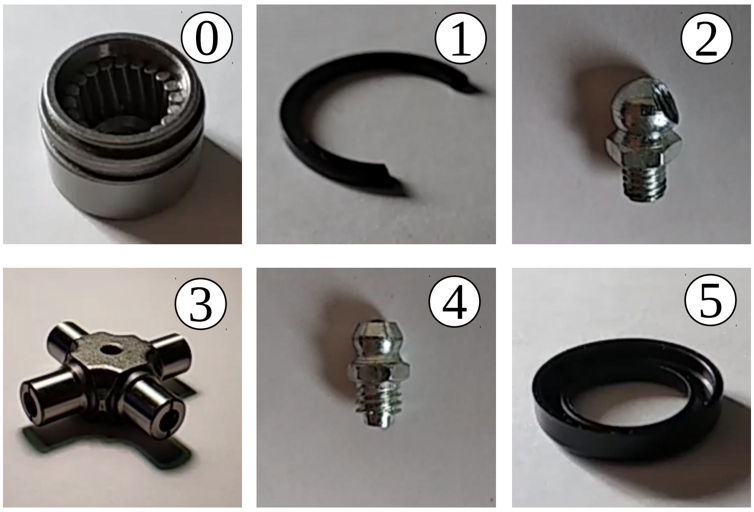


**Figure 5.4:** *Samples of the parts of the crosshead gimbal used as a target domain with the index of the experiments. The variation of the backgrounds were minimized using a white background.*

dielectric parts (Figure 5.4). Hence the distribution of some of its characteristics is known beforehand.

The photos of the gimbal were obtained recording videos around each one of the objects in a white background and sampling from the frames a defined number of images. Additionally, 20% of the samples of each class were objects with some kind of extrinsic variations (grime, corrosion among others). Thus, the photos will be taken controlling as much as possible the intrinsic and environmental variables to study the effect of the extrinsic variation. Thus the controlled variables are:

- Geometry: the exact 3D model of the part is known and there will not be intra-class variations
- Material of the parts: all the parts will have the same material across all the samples
- Background of the parts: plain white color
- Number of samples per class
- External elements, there will be not external elements. In each sample only will be a target class.
- Extrinsic variations (grime and corrosion)

The photos dataset is divided in two, 70% for training as if it was a regular ML problem and 30% for testing. This testing dataset is used to evaluate the performance of each source dataset compared. In the experiments, the CNN models were trained with three source datasets: proposed synthetic method (Chapter 4), Clay rendering and the real photos.

For each one of the methods was trained 8 acrshortcnn models using two approaches. (i) fine tuning a pre-trained model in *ImageNet* with the source datasets. (ii) Trained the models from random initialization, where all of the layers of the model are updated in the training process. The experiment was conducted several times ($8$ for each source method) in order to have a statistical approach given the variations due to the random process of CNN training.

For this experiment we used Mobilenet with the next parameters: Dropout $0.8$, $\alpha = 1$, $\rho = 1$ and samples resolution $224$px. For the fine tuning, the models were trained $10$ epochs and for the random initialization $20$ epochs

Finally for each trained model, is calculated the accuracy per class. That is obtained by measuring the prediction of the model vs the ground truth when passing the images of each class of the photos testing dataset.

### 5.3.2 Unknown characteristics

This set of experiments were made with a target domain with *unknown characteristics* and their distribution. Where the only information is known is the general category of the class name (e.g nuts, screwdrivers). There was not known the specific geometry of the parts, materials or environments. The used target dataset is a subgroup of $28$ classes (Appendix 5.8.1) obtained from *ImageNet* [191].

The main experiments performed for unknown domains are summarized in Table 5.1. Additionally, for each main experiment other variables were considered such as blurring the images, the number of training epochs or changes in the target dataset that will explain in deep in each experiment.

**Table 5.1:** *Experiments summary with real dataset with unknown characteristics. List values indicates that the experiment were conducted with each one of the values. The values of the last column (exp, fixed and rand) make reference to three types of configuration used to create the synthetic datasets . Where the parameters of the DV variables are drawn from specified the PDF*

| Exp. | Models | Classes | Resolution | Fine-tune | Epoch | Configuration |
|---|---|---|---|---|---|---|
| 1 | 25 | 5 | 224 | no | 10 | [exp, fixed, rand, clay] |
| 2 | 25 | 5 | 224 | yes | 5 | [exp, fixed, rand, clay] |
| 3 | 5 | 10 | [64,128,256] | no | 10 | exp |
| 4 | 5 | [5,10,20] | 224 | yes | 5 | exp |

In the Table 5.1, the lists of values indicates that the experiment was conducted with each one of the values. The last columns refer to the method used for creating the synthetic dataset. As the method proposed for the synthetic generation allows to define the parameters of the variables (light intensity, POV, amount of corrosion among others) from Probability Density Function (PDF). Different configurations used for the creating of the synthetic datasets:

**Exp** Intended to produce a smooth distribution of the parameters from an expected value. The most relevant parameters are generated with the next configuration, full configuration is found in Appendix 4.9.1:

- POV: Gaussian with rotation: $\sigma = 40, \mu = [0, 0, 0]$, in this case is the typically expected view that was acquired from the most common view angle of the 3D models, the rotation is the Euler rotation of the 3D model. For the position:

$\sigma = [800, 0, 0], \mu = 100$ where the camera is located at $(0, 0, 0)$ and pointing to the positive $x$ axis.

- Lights intensity: temperature and direction: random
- Environment lights: Gaussian $\sigma = 0.2, \mu = 2.5$
- Number of external elements: random pick of $[0, 1, 5]$ number of elements
- Material probability: 50% metal
- Intrinsic variables: negative exponential distribution (Corrosion $\beta = 0.1$, grime $\beta = 0.1$, superficial imperfections $\beta = 0.2$ and micro-displacements $\beta = 0.2$)

**Fixed** : the extrinsic variables to be $0$ and no external elements

**Rand** : all the variables are set to random without external elements

Additionally, a state of the art method for rendering called **Clay rendering** was used to compare the use of PBS versus a not realistic approach for creating datasets.

Finally, in the fine-tuning process, a pre-trained model is loaded. In this case, we choose to use a pre-trained model in *ImageNet*, and in this experiments do not use the classes in which the model was pre-trained. In total $28$ were used for this experiment from a total of $45$ classes that were used for initial experimentation. The process consists in load the pre-trained model, set all the convolutional layers weights to be fixed (not trained) and the weights at the top layers, the fully connected, are trained with the new dataset.

## 5.4 Generation of Datasets

### 5.4.1 Synthetic dataset

The generation of the synthetic datasets is performed in two steps: First is to collect the next assets:

**3D models** In the case of the *unknown characteristics* the 3D models from both target classes and external models were download from *3D Warehouse* [1] that is a free online repository of 3D models. Different 3D models from each class were downloaded in DAE format and prepared using the algorithm presented in Section 4.7.3. In total $2044$ models were downloaded and fixed for the target elements and for the external elements a total of $306$ models.

**Intrinsec Variation Maps** The variation maps were also available in different web-pages [5–7]. Additionally for them to work better, they were processed for make them "seamless". The algorithm used is the one available in *Gimp* [4] that blend the opposite content of the borders, allowing to avoid strong variations of the texture when it repeats over the surfaces.

**Light probes** Similarly light probes are available online in multiple websites. Some of the used from the experiments were downloaded from [54].

Next step is to define the type of distribution of the creating a JSON configuration file with the variables shown in Appendix 4.9.1. Additionally, this configuration file defines another property of the renders, the ones used for the experiment were:

- Resolution: 128px for unknown and 224px for *known characteristics* experiments
- Sampling: 10
- Number of renders per class: 1000

Some examples of the renders generated for the experiments by the proposed method are shown in Figure 5.5. These examples were taken from the dataset generated using the exponential distribution configuration (Appendix 4.9.1). In the Figure are shown the classes (a) screw, (b) nut, (c) piston, and (d) gear.

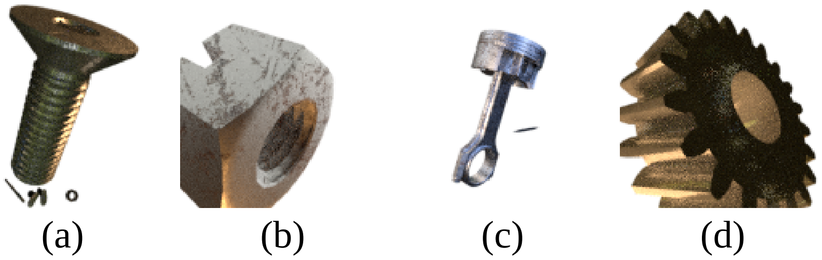


**Figure 5.5:** *Examples taken from the dataset generated by the proposed method using the exponential distribution configuration (Appendix 4.9.1). In the Figure are shown the classes (a) screw, (b) nut, (c) piston and (d) gear.*

The examples of the samples generated with the Clay method are presented in Figure 5.6

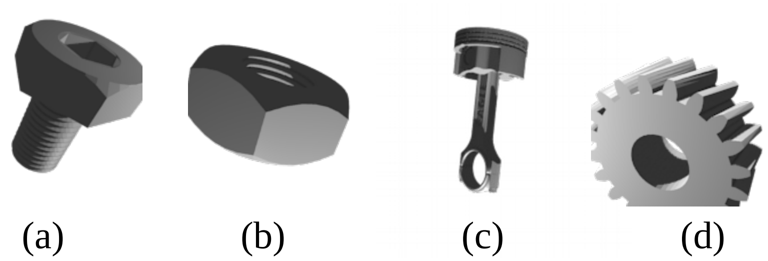


**Figure 5.6:** *Clay rendering examples taken from the dataset used for the experiments. In the Figure are shown the classes (a) screw, (b) nut, (c) piston and (d) gear.*

### 5.4.2 Real Photos Datasets

The acquisition of the real images, as mentioned in the previous (Section 5.3), two real photos datasets were acquired. The first one a subset of *ImageNet* where each class is access through an ID (The ID for the experiment are available in Section 5.8.1) and were downloaded using [203].

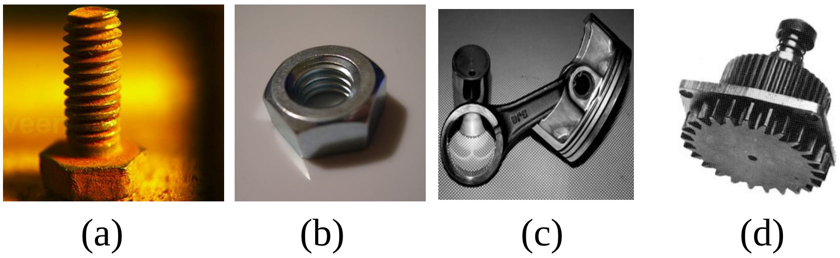


**Figure 5.7:** *Example taken from the ImageNet dataset. In the Figure are shown the classes (a) screw, (b) nut, (c) piston and (d) gear.*

The example of images taken from the *ImageNet* dataset are showed in Figure 5.7

For the *known characteristics* dataset, the photos were taken. For this study, 6 industrial parts of a crosshead gimbal were selected. For each one of the parts, different videos of the parts in different positions were recorded. The backgrounds of all the parts were the same, a white paper (Figure 5.4).

Additionally, the 20% of the samples of the gimbal dataset were composed of parts with some kind of extrinsic variation (Section 4.6). In order to achieve this, different parts were used and the samples were extracted from videos of the parts. For our experiments, we sampled 800 frames of the video of parts without [1] extrinsic variations and 200 from the videos of objects with this variations.

## 5.5 Results

In this section are presented the results of the experiments for each type of dataset. The metric used to measure the closeness of a prediction of the true value is the accuracy $A_{cc}$ defined in Equation 5.1. Where $T_p$ are true positive, $T_n$ true negative and $m$ the number of samples.

$$A_{cc} = \frac{T_p + T_n}{m} \tag{5.1}$$

### 5.5.1 Known Characteristics Dataset Results

**Fine tuning**

In this section are presented the results of fine tuning a pre-trained model in *ImageNet* to a specific domain (Section 5.3.1) using three different datasets. (i) Synthetic: the method proposed in Chapter 4. (ii) Clay rendering and (iii) using real photos.

The process followed was to train only the fully connected layer of the pre-trained model with each one of the datasets and evaluate how well they predict the appearance of the classes in the images of the photos test dataset. The general results are shown in Table 5.2.

[1] Some variations as surface imperfections such as smudges or small scratches are always present in real life objects

**Table 5.2:** *Known statistic experiment results summary of fine tuning models with source datasets (Synth, Clay and Photos) in the task of classification in the domain of the Photos. The distribution of the accuracies per class were compared with the ones produced with the Synth training using Levene and Monte Carlos randomization tests*

| Source | Acc.(%) | Levene pval ( vs Synth) | MC rand. Pval (vs Synth) |
|---|---|---|---|
| Synth | 53.2 | - | - |
| Clay | 10.9 | 5.90E-06 | 0 |
| Photos | 75.6 | 0.005 | 0.014 |

In the results are compared the distribution of the prediction accuracies per class of the models trained with the Synth method versus the Photos and Clay trained models. None of the distributions were Normal, thus a no parametric analysis was performed. Nevertheless, the distribution of the accuracies seems to have a significance (p-value < 0.05) difference in the variance (Levene test). Therefore a mean analysis was performed using a Monte Carlo Randomization [84] showing that the compared samples seem to be drawn from the distribution with different means (Table 5.2).

The distribution of the accuracies per class of the models trained with Synth (red) and Photos (blue) sources are shown in Figure 5.8. Further, the best accuracy model for source training is shown in Table 5.3.

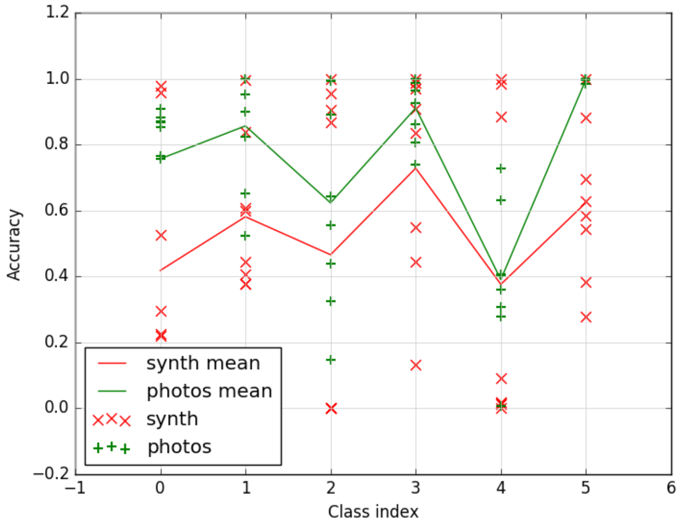


**Figure 5.8:** *Accuracy distribution per class of predictions of the models fine tuned with Synth (red) and Photos (green) datasets. The models predict the presence of a class in an photo of a target testing dataset*

Additionally, as the synthetic images are saved with the alpha channel, allows to variates its background. Thus a comparison was made between augmenting the synthetic dataset with random backgrounds or using the same as the target dataset. The results showed no significant difference (Levene p-val: $0.64$ and Kruskal–Wallis p-val: $0.5$).

The models trained with the same background that the target obtained an accuracy mean $6.4\%$ better than the random backgrounds. Similarly, the best accuracy obtained by

**Table 5.3:** *Best accuracy obtained by one of the models fine tuned with the evaluated source datasets*

| Source | Mean | Class idx | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 |
| Synth | 66.8 | 52.5 | 99.7 | 0 | 90.7 | 88.6 | 69.5 |
| Clay | 17.2 | 0 | 0 | 89.6 | 13.3 | 0 | 0 |
| Photos | 85.6 | 90.7 | 100 | 55.4 | 96.3 | 72.7 | 98.4 |

the models of the random backgrounds was $7.1\%$ inferior.

#### Random initialization

In this case, all the models were trained from initial random values as well as all the layers weight were updated during the training. The summary of the results are shown in Table 5.4 where is shown the mean accuracy of prediction of the models over the target domain by each source training.

**Table 5.4:** *Known statistic experiment results summary of comparing three source of training datasets (Synth, Clay and Photos) in the task of classification in the domain of the Photos. The distribution of the accuracies per class were compared with the ones produced with the Synth training using Levene and Kruskal tests*

| Source | Acc.(%) | Levene pval ( vs Synth) | Kruskal-pval ( vs Synth ) |
|---|---|---|---|
| Synth | 23,6 | - | - |
| Clay | 14,6 | 0.18 | 0.031 |
| Photos | 64.9 | 0.28 | 7.04E-08 |

Further, (Table 5.4) present the results of the homoscedasticity test (Levene) and variance analysis (Kruskal–Wallis) comparing the Photos and Clay methods with the Synthetic. The best accuracy for each training type is found in Table 5.5.

**Table 5.5:** *Best accuracy obtained by one of the models trained with the evaluated source datasets*

| Source | Mean | Classes Index | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 |
| Synth | 28.5 | 15.6 | 14 | 0 | 80 | 62 | 0 |
| Clay | 17.2 | 0 | 97 | 0 | 0 | 0 | 5.6 |
| Photos | 93.4 | 99.7 | 87.8 | 95.7 | 96.7 | 80.1 | 100 |

The distribution of the accuracies per class is illustrated in Figure 5.9 comparing the Synth (red) and Photos (green) predictions.

Comparing the models trained with the photos versus the models fine-tuned by Synth, there was not a statistical difference between the accuracies achieved by both (Levene p-val: 0.48 and Kruskal–Wallis p-val: 0.11). Where the models trained with the photos obtained a $11.7\%$ more in the average than the fine-tuned synthetically.

On contrary, fine tuned models by Clay source presented on average a lower performance (3.7% of difference) than their counterpart trained from random initialization. This difference was not significant under the Levene (p-val: $0.53$) and Kruskal–Wallis (p-val: $0.61$) tests.

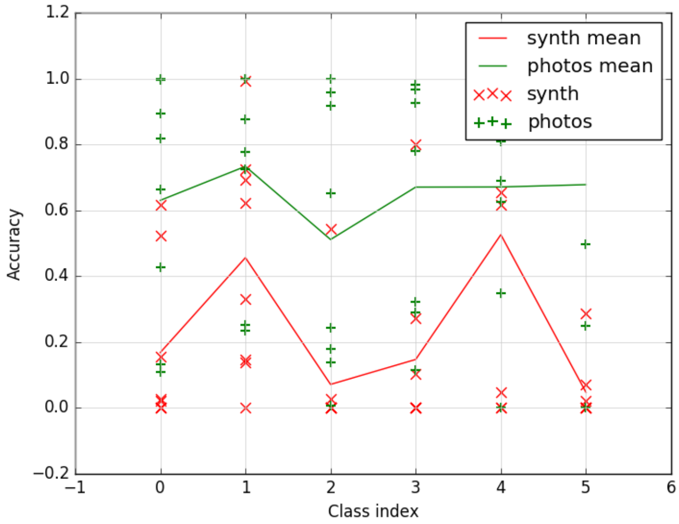


**Figure 5.9:** *Accuracy distribution per class of predictions of the models trained with Synth (red) and Photos (green) datasets. The models predict the presence of a class in an photo of a target testing dataset*

Additionally, comparing the results of the models that were fine tuned by the proposed method with the results obtained training with photos from random initialization, no significant difference where found (Levene p-val: 0.48, Kruskal p-val: 0.11). The comparison of the distribution of the accuracies is shown in Figure 5.10.

### 5.5.2 Unknown Characteristics Dataset Results

**Experiment 1**

In this experiment 25 models of 5 classes randomly picked and equal for all the evaluated methods (exp, random, fixed, clay and photos) are trained from a random initialization and all weights of the layers are updated during the back-propagation. Later the accuracy of prediction of the target dataset for each class is calculated.

This is performed by passing each class of the target dataset for the trained model an calculating the accuracy of the prediction (Equation 5.1). The statistical comparison of the mean accuracy of the predictions of the target domain between the different sources for training the CNN models from random initialization is shown in Table 5.6.

Further, for minimizing the noise in the comparison, a "clean" evaluation test was also used (Acc-clean in Table 5.6) where samples that were not belonging to the proposed problem were removed. Therefore samples with more than two target classes, or that were miss labeled were removed.

Additionally, the accuracies from all the sources presented the same variance Levene Test (p-val>0.05). But a significant difference between the medians (Kruskal–Wallis test) between the proposed method (in all the configurations) and the Clay and using real images from the domain (imgnet) in Table 5.6. The comparison of the mean accuracies per class of the sources can be seen in Figure 5.11.

Similarly the exponential configuration present a statistical difference with the random

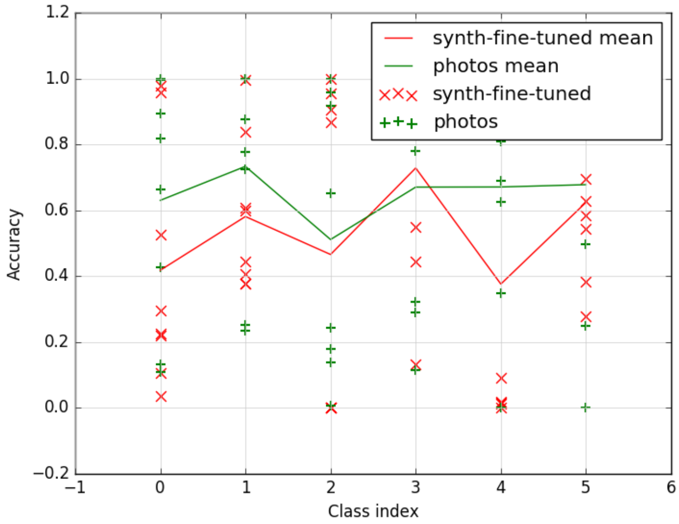


**Figure 5.10:** *Accuracy distribution per class of predictions of the models fine tuned with Synth (red) and models trained with Photos (green) from random initialization.*

**Table 5.6:** *Experiment 1 of unknown characteristics summary. Statistical comparison between the different sources for generating synthetic datasets (except imgnet). The accuracy (Acc) is the mean of the prediction accuracies of the models trained with the source evaluated with real photos from the testing set of ImageNet. For the Acc-clean The ImageNet test dataset was cleaned of incorrect samples. The variance of all the treats were equal according to the Levene Test (p-val>0.05).*

| Source | Acc(%) | Acc-clean(%) | Kruskal (pval<0.05) |
|---|---|---|---|
| Exp | 23,9 | 26,3 | Clay, Imgnet |
| Fixed | 24,3 | 26,8 | Clay, Imgnet |
| Rand | 21,7 | 21,6 | Clay, Imgnet |
| Clay | 20,3 | 20,5 | All |
| Imgnet | 40 | 43,2 | All |

models in the target domain of the original ImageNet (Kruskal p-val: 0.03). Also the random configuration presented significant difference (Kruskal p-val: 0.01) with the fixed model regarding the clean ImageNet.

An example of the distribution of the accuracies per class trained with the synthetic dataset with a fixed configuration can be seen in Figure 5.12. Where each point represents the accuracy of prediction of a model regarding one class.

### Experiment 2

In this experiment 25 models were trained for each source dataset (exp, random, fixed, clay and real photos) and all the models were fine-tuned using a pre-trained *ImageNet* model whose classes were not the same as the ones considered in the experiment.

The fine-tuned method consists of load the pre-trained model and freeze the weights of the convolutional layers at training time and only update the fully connected layers. The fully connected layers are randomly initialized. These random initial values are the same

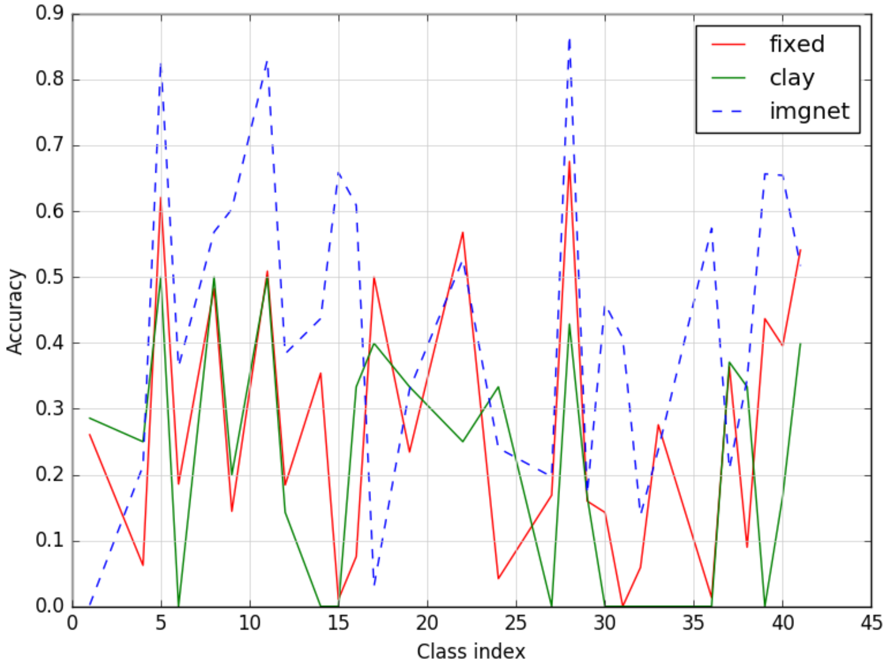


**Figure 5.11:** *Mean of accuracies per class of models trained with source datasets (fixed, clay, imgnet). Each model was trained in the same conditions with random initialization. A total of 25 models of 5 classes for each source was evaluated*

across the methods together with same classes at training time. For each new comparison, new random classes and initial values are computed.

Afterward, the accuracy of the prediction for class of each model is performed. This is achieved by passing each class of the dataset for the models. In the Table 5.7 are summarized the results of the mean accuracies of the prediction of the models. Similarly, the models were also evaluated in a "cleaned" version of *ImageNet*.

In the case of fine-tuning, no significant difference between the median (Kruskal test pval>0.05) and variance of the accuracies was found in the synthetic methods. In contrast, all the methods were found to have a significant p-value for the Levene test (pval>0.05), thus all the source groups accuracies seem to be originated from populations with the same variance. In Figure 5.13 are shown the means of the accuracy of models for each

**Table 5.7:** *Summary of mean accuracies achieved by models fine tuned using different source datasets. All the methods were found to have a significant p-value for the Levene test (pval>0.05)*

| Source | Acc(%) | Acc-clean(%) | Kruskal (pval<0.05) |
|---|---|---|---|
| Exp | 51,4 | 55,1 | - |
| Fixed | 50,4 | 53,8 | - |
| Random | 47,9 | 50,4 | - |
| Clay | 49,5 | 52,5 | - |
| Imagenet | 66 | 71,2 | all |

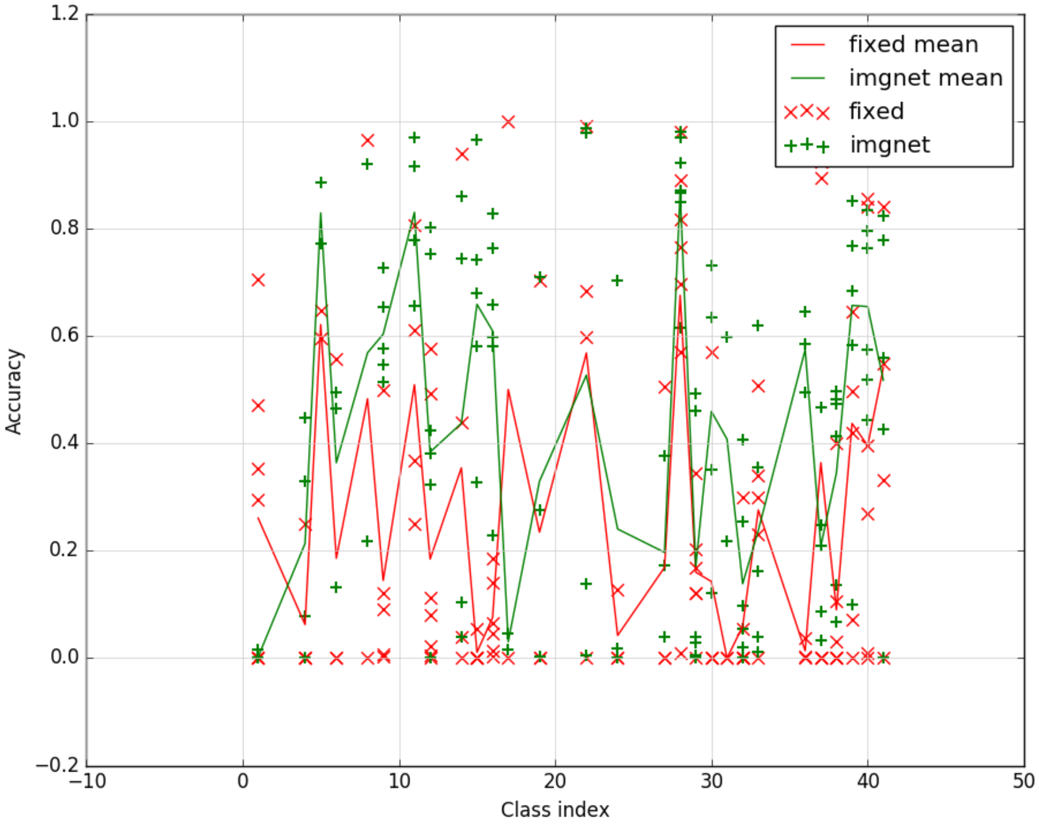


**Figure 5.12:** *Accuracy of ImageNet prediction for class of models trained with ImageNet (Green) and synthetic-fixed (red) datasets. In total* $50$ *models were trained with random classes equal for each type of training. Similarly same initial values were set for the random initialization*

source method.

Additionally, in Figure 5.14 is shown the accuracy distribution of predictions per class of the fine-tuned models trained with *ImageNet* (green) and synthetic dataset generated with the exponential configuration (red).

Further, another test was performed regarding the evolution of the models during the training time. In this test was compared the accuracies of the prediction of the clay and exponential method after the models being trained during $[5, 10, 15, 20, 25]$ epochs. The results are shown in Table 5.8.

An additional test was performed a random blurring filtering operation in the input images of the synthetic (exp) dataset and the results show an underperform (blur 50.7% vs no blur 51.4%) without a significant difference.

Compared with the results of using models with random initialization, all the methods obtained a boost in their performance (exp $+28.8\%$, random $+28.8\%$, fixed $+27\%$, clay $+32\%$, imgnet $+28\%$).

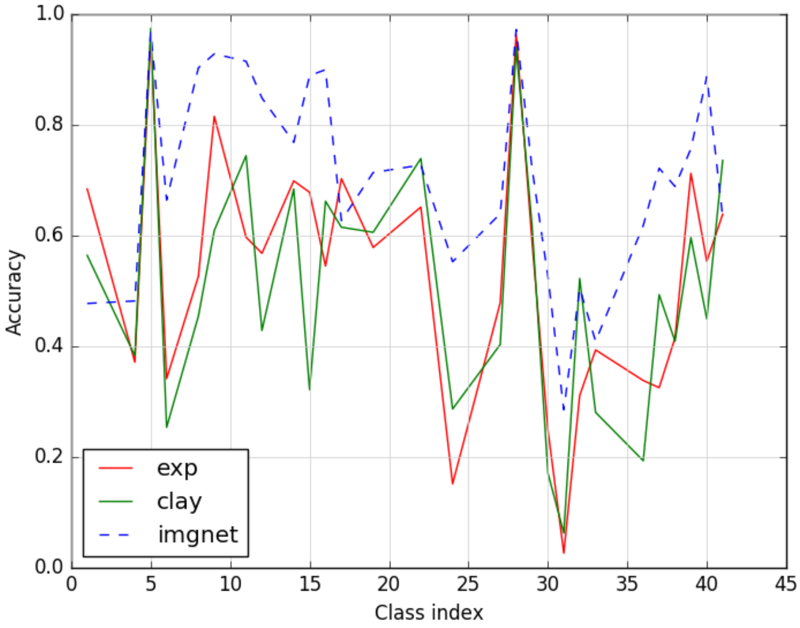


**Figure 5.13:** *Mean of accuracies per class of models fine tuned with source datasets (exp, clay, imgnet). Each model fine tuned a pretrained model of ImageNet. A total of 25 models of 5 classes for each source was evaluated*

**Table 5.8:** *Mean accuracy evolution on target domain prediction after* $[5, 10, 15, 20, 25]$ *epochs of training using exponential and clay datasets*

| Epochs | Exp (acc. %) | Clay (acc. %) |
|---|---|---|
| 5 | 55,1 | 52,5 |
| 10 | 56,3 | 52,9 |
| 15 | 56,5 | 54,5 |
| 20 | 57,9 | 55,2 |
| 25 | 58,1 | 54,1 |

### Experiment 3

The experiments $3$ is aimed to show the influence of resolution of the training images in types of source datasets. The resolutions (in pixels) investigated are $[64, 128, 224]$. Therefore the input images were all scaled up to the defined resolution before the training process.

All the models were initialized from random values and trained for $10$ epochs. Equally, the models were trained for the visual problem of recognizing if an object is present in an image from a total of 5 object categories. The configuration used to generate the synthetic dataset was *exp*.

The summary of the results is shown in table 5.9. Where is shown an increasing difference between the accuracies once the resolution of the images increases as well as the difference in the distribution of the accuracies according to the statistic tests (Kruskal and Levene) between the compared types of training.

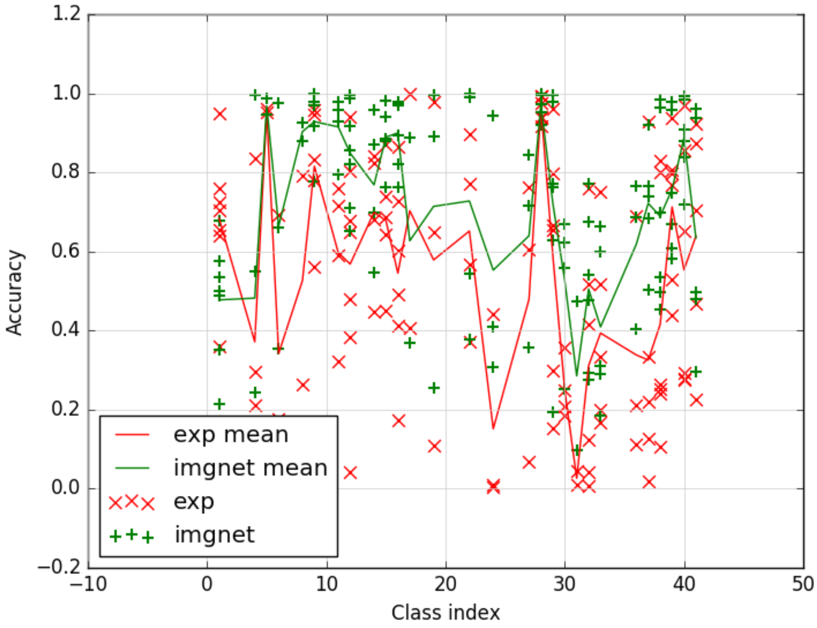


**Figure 5.14:** *Accuracy of ImageNet prediction for class of models fine tuned with ImageNet (Green) and synthetic-exp (red) datasets. In total* 50 *models fine tuned a pretrained model of ImageNet.*

**Table 5.9:** *Summary of the effects of training with different sample resolutions* $[64, 128, 256]$ *in the synthetic-exp and real photos (imgnet) training for domain adaptation*

| Resolution (px) | Exp (acc. %) | Imgnet (acc. %) | Acc. Diff. (%) | Levene (pval) | Kruskal (pval) |
|---|---|---|---|---|---|
| 64 | 23,7 | 32,6 | 8,8 | 0,28 | 0,06 |
| 128 | 24,9 | 37,5 | 12,6 | 0,28 | 8E-03 |
| 256 | 23,6 | 41,5 | 17,8 | 0,5 | 1.20E-03 |

The relationship between increasing the resolution and increasing the difference between the accuracies of the methods can be seen in Figure 5.15. Where is shown the accuracy difference between the two methods per class in the 64px (blue) and 256px (orange). Positive values indicate that the accuracy increased in the synthetic training. In total, the difference increase a 9% from 64px to 256px.

#### Experiment 4

Investigate the effect of the number of classes in fine-tuned models with synthetic and real datasets. The resolution used (224px) is derived from the one used in the proposed article of the architecture used (MobileNet). The number of classes evaluated was $[5, 10, 20]$, the models were trained for 5 epochs and 5 models for each number of classes.

The summary of the results shown in Table 5.10. Is shown the decrease in accuracy in both models once the number of classes increases, and it is more aggressive in the photos than in the synthetic dataset. Further, no significant difference (Kruskal–Wallis and Levene tests) in the accuracies were found in the models of the 5 and 20 classes.

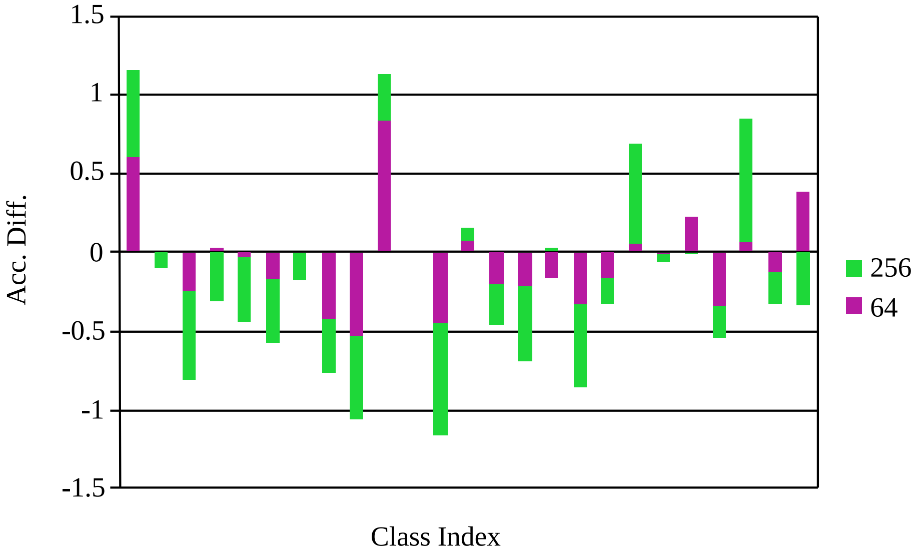


**Figure 5.15:** *Accuracy difference between training with real photos (imgnet) and synthetic-exp. In the x-axis are the evaluated class index and in y-axis the accuracy difference. Positive values indicates that the accuracy increased in the synthetic training.*

**Table 5.10:** *Summary of mean accuracies achieved by models fine tuned with synthetic and target photos variating the number of classes to classify.*

| No. Classes | Exp (acc. %) | Imgnet(acc. %) | Acc. Diff (%) | Exp Acc. Decr.(%) | Imgnet Acc. Decr.(%) |
|---|---|---|---|---|---|
| 5 | 48.7 | 62.4 | 13.7 | 0 | 0 |
| 10 | 26.7 | 47.4 | 20.7 | 22 | 15 |
| 20 | 20.9 | 34.3 | 13.4 | 5.8 | 13.1 |

Additionally in Figure 5.16 is plotted the mean accuracy per class of the two sources, imgnet (left) and synthetic-exp (right) with the different learned number of classes (5-blue, 10-orange, and 20-yellow)

Finally a comparison of the average accuracy obtained by the models between the Known and Unknown experiments is showed in Table 5.11. Likewise is compared the difference between models fine tuned and trained from random initialization.

**Table 5.11:** *Comparison of the average accuracy obtained by the models between the Known and Unknown experiments. Likewise is compared the difference between models fine tuned and trained from random initialization.*

| | Random Initialization | | | Fine Tuning | | |
|---|---|---|---|---|---|---|
| | Clay | Synth | Photos | Clay | Synth | Photos |
| Unknown | 20.3 | 24.3 | 40 | 49.5 | 50.4 | 66 |
| Known | 14.6 | 23.6 | 64.9 | 10.9 | 53.2 | 75.6 |
| Diff. | -5.7 | -0.7 | 24.9 | -38.6 | 2.8 | 9.6 |

The relation ship between the average accuracies in the Known (orange) and Unknown (blue) experiments of the different source training (Clay, Synth and Photos) is shown in Figure 5.17. On left the models trained with source from random initialization. On right

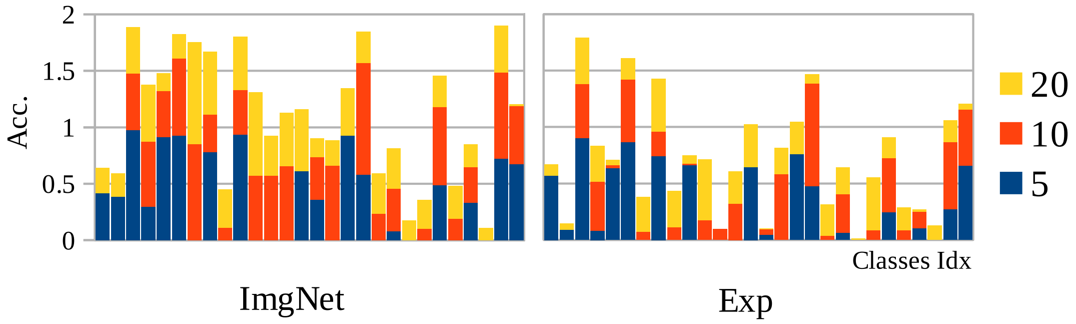


**Figure 5.16:** *Mean accuracy per class of the two sources, imgnet (left) and synthetic-exp (right) with the different learned number of classes (5-blue, 10-orange, and 20-yellow).*

models fine tuned with source datasets. These plots show a kind of symmetry regarding the Synth dataset, a large difference between models fine tuned with Clay dataset. Similar to a large difference between the models trained with Photos in the random initialization.

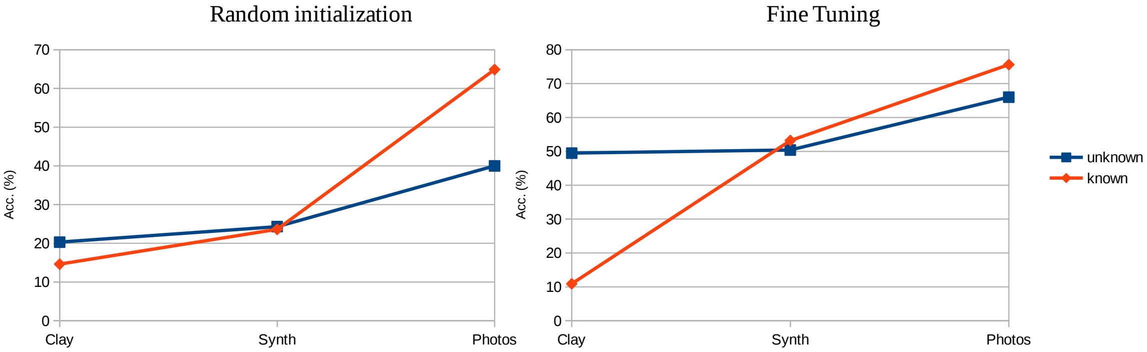


**Figure 5.17:** *The relation ship between the average accuracies in the Known (orange) and Unknown (blue) experiments of the different source training (Clay, Synth and Photos). On right the models trained with source from random initialization. On left models fine tuned with source datasets.*

Additionally, the accuracy difference between the models in the experiments of known and unknown characteristics is showed in Figure 5.18 (Last row of Table 5.11). Represent the changes between a known small domain and a broader domain.

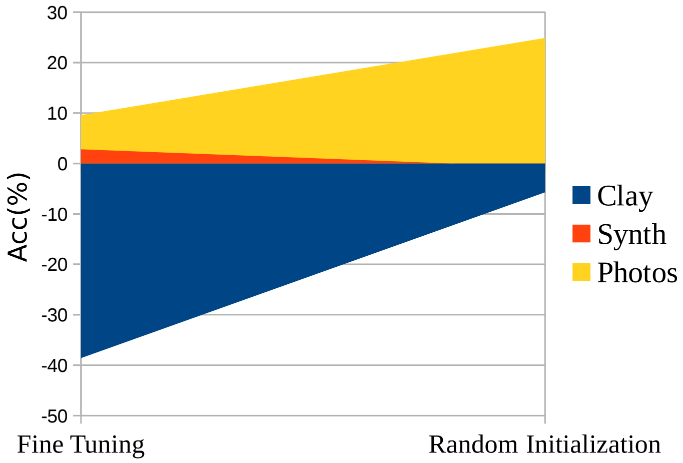


**Figure 5.18:** *Accuracy difference between the models trained and fine tuned with the different sources (Clay, Synth and Photos) under known and unknown characteristics experiments.*

## 5.6 Discussion

According to literature, increasing the level of realism in synthetic training datasets increase the performance of prediction over the target domain [160,178,251]. Hence models trained with shaded renders could surpass the performance of models trained with wire-frame renders. Have been proved that using synthetic data created with PBS and realistic lighting improves the overall performance of surrounding training methods.

Following that line of thought, the next step is to recreate the variations present in real-life given that it is not enough with using physically accurate light-matter interaction models that represent perfect materials if in reality objects are rarely or almost never found in perfect conditions.

Additionally, of having physically based models that recreate light-matter with real-life imperfections, the characteristics of the synthetic objects and environment needs to be distributed with the same bias that the target ones.

The results of this research support these line of thoughts, given that there is a significant difference in models trained with PBS approach than using clay rendering supporting the idea that increasing realism boost the performance of the models.

Further, recreate the high-level bias (e.g POV or level of imperfections) equally affects the performance in predicting on the target dataset. This is illustrated by the results of the synthetic method that using different distributions (exp, fixed and random) reach statistical different performances (Table 5.6).

Nevertheless, fine-tuning models that have been pre-trained in larger datasets that have similar classes, can help to blur this differences (realism and bias). In the results of Experiment 5.5.2, models were pre-trained in a different domain but it was linked to the target domain as both were part of *ImageNet*. The results showed that there were no significant differences in the means of realistic rendering versus clay rendering.

Such results support the research of [177] that realistic low-level features are not required to be simulated when models are pre-trained in a similar domain.

Several reasons could explain this results, first, the similarity in target datasets and pre-trained datasets (both subgroups of *ImageNet*) overshadow the difference between the synthetic datasets. That there were many unknown variables (e.g. models, backgrounds, lights) and the bias of the datasets overcome the difference in the realism of the source datasets.

Therefore a difference in marginal distribution between the datasets were larger than the difference in the recreated features, making less noticeable the difference between the datasets.

However, when the target domain and the pre-trained model domain are different (Known Experiments 5.5.1). The results show that there is a significant difference in using the proposed method versus the Clay rendering in both fine tunings (Table 5.2) and random initialization (Table 5.4).

On the other hand, the fine-tuned models with the Clay dataset showed on average a larger difference (38.6%) between the known and unknown experiments (Table 5.11). This is illustrated in Figure 5.17 on the right image. Showing a large difference in the accuracies on the models fine-tuned with the Clay datasets.

This could be due to that the source dataset and target datasets are unrelated, thus the training could tune the models to over-fit the incorrect features, non-realistic present in the Clay dataset. But when the characteristics are unknown more variability in the datasets avoid the over-fitting.

On the contrary, models trained from random initialization with photos in the *known characteristics* achieve a higher accuracy given that they over-fit to the specific case with the correct features. This is illustrated in Figure 5.17.

Worth noting the particular inverse symmetry that this two cases around the Synth source dataset (Figure 5.17). Where the difference of the accuracies between known and unknown experiments are minimal (Table 5.11). This shows that the features of the proposed method (Synth) are not as close to the bias of the target but neither as far as the Clay dataset.

Where the accuracy difference of the models trained with the sources (Clay and Photos) under known and unknown experiments shown an inverse behavior of fine tuning and training from random initialization (Table 5.11).

Additionally, models trained from random initialization using the synthetic data (Clay and Synth) obtained lower performance in both experiments (Known and Unknown). For instance, the proposed method (Synth) in Known (26.3%) and *unknown characteristics* (23.6%) (Figure 5.17).This could be interpreted as the variety of the features of objects found in the training set is not as large as the ones found in the testing set which also explain the better performance of the fine-tuned models.

Regarding the results of the experiment of the effect of the resolutions (Section 5.5.2). The results showed that using the photos the as the resolution increase, the performance increase. Nonetheless, regarding the synthetic data the performance clips at 128 pixels that is the resolution of what the images were produced. These results suggest that the details provided by a higher resolution in both synthetic help to increase the performance. However when the training images are scaled the same amount of details prevail.

In all the experiments of the *unknown characteristics*, is shown how the performance variates greatly with respect to the chosen classes. And that more or less the classes accuracies behaves with the same pattern across the different experiments (*unknown characteristics*). This suggests that the accuracy depends on the differentiability between among the selected classes and the background. And that this differentiability broadly is shared among the different types of representation (realistic, clay or photography).

This can be seen in the pattern formed by the mean accuracy of fine-tuned models comparison in Figure 5.13.

Finally, fine tuning models with the proposed method when the characteristics achieved comparable results (no statistic difference) with models that were trained with photos. This suggest that the proposed method could be considered as an alternative for training models for the object recognition task.

This task is fundamental in the development of AR applications, as it allows to link the real and virtual elements. Besides as the method proposed is able to generate another type of

Additionally, the proposed method can be used together with another DA in order to increase the performance. Other alternatives are mixing with real photos that could be from or outside the domain and hyper-parameters exploration regarding the specific case.

## 5.7 Conclusions

In this Chapter have been performed and ablation study related to the use of the proposed method for the creation of synthetic datasets that simulates with a PBS approach variations present in the domain of IAR.

The performed study included two types of experiments, one with *unknown characteristics* about the target domain and other with *known characteristics* meaning that some of the characteristics of their parts were known. The conclusion found in these set of experiments were:

- Fine tuning models with proposed method reach comparable (No statistical difference) with models trained with photos with *known characteristics* (Figure 5.10). These results validate the proposed method as a viable alternative for training surrounding understanding algorithms applied to industrial cases.

- H1. Results suggest that the models trained with datasets generated with the proposed method have a significant difference compared to models trained with the datasets generated without taking into account realistic light-matter interaction when the source and target domains bias are similar.

- H2. Similar patterns of performances were found across different training types (synthetic and photos). Further, in almost all of the cases, no statistical difference was found in the variance of the accuracies between the source methods. This suggests that across the methods a comparable behavior occurs related with the classes considered.

- H3. The result showed a statistical difference between the accuracies achieved by models trained with synthetic and source domains at different resolutions. Further, this difference of accuracies increases in a proportional relation with the resolution of the input data.

- H4. Fine tuning models with synthetic data showed significant changes in the performance of prediction of the target domain compared with models trained from random initialization. The results suggest that the changes depend on the similarity with the target domain. Therefore this hypothesis is rejected as performance could decrease if the features of the synthetic dataset mislead the training regarding the target (Table 5.11).

- H5. When the target and source domain are not highly related, the increase of the performance in the source results in a decrease in the target as the model is fitted to another domain. On contrary highly related domains when the model is fitted to the source it is performed equally in the target (Figure 5.17).
- H6. Datasets produced with different characteristics (random, exponential and fixed) using the proposed method (Physically based) present significant difference in the performance (Experiment 5.5.2). This difference could be attributed only to the bias of the datasets.
- Is equally important to recreate the feature space (simulate the same characteristics) than using the same marginal probability distribution of the target dataset.
- Models trained with realistic rendering but substantial difference in the marginal distribution with the target underperformed equally than models trained with non-realistic rendering.
- Fine tuning models pre-trained in the same domain make irrelevant the recreation of realistic low-level features when there is a considerable difference of the marginal distribution of the source and target domains.
- There is an increment of the average accuracy in models trained and fine-tuned that is approximately proportional to the realism of the source datasets. The slope of this increment is approximately proportional to the similitude between the marginal distributions of source and target datasets.
- The results found support the use of the proposed method as valid option to train CNN models in the vision task of image classification. Where its use could reduce the burden of implementation and development of IAR applications. By improvements of the current method, it could be applied to other cases where the accuracy is more significant.

## 5.8 Appendix

### 5.8.1 ImageNet Classes

List of ImageNet classes and experiments id.

```
{
   "nut": 24,
   "pen": 10,
   "shock": 39,
   "key": 31,
   "telephone": 42,
   "washer": 29,
   "toilet": 2,
   "drill": 17,
   "pliers": 36,
   "bottle": 7,
   "sprocket": 14,
   "carabiner": 27,
   "hinge": 38,
   "gear": 41,
   "mouse": 23,
   "lock": 3,
   "joint": 12,
   "screwdriver": 18,
   "flashlight": 9,
   "bearing": 5,
   "lamp": 25,
   "outlet": 6,
   "chair": 21,
   "gun": 0,
   "oring": 28,
   "caster": 37,
   "propeller": 30,
   "valve": 40,
   "pulley": 32,
   "hammer": 34,
   "screw": 20,
   "spoon": 35,
   "faucet": 8,
   "rim": 22,
   "scissors": 15,
   "relay": 16,
   "microphone": 26,
   "clamp": 19,
   "cup": 44,
   "pump": 11,
   "clock": 13,
   "pin": 1,
   "camera": 43,
   "spring": 33,
   "piston": 4
}
```

# CHAPTER 6

# Concluding remarks

This research was focused on the analysis and development of methods for easy the implementation of Augmented Reality (AR) applications in industry. This problem was faced by two sides, developing a global framework that presents different variables that are required to be taken into account at the time of developing new Industrial Augmented Reality (IAR) applications.

This framework could be used to give developers or any researcher with interest in the field key elements for implementing AR systems. Also to provide a list of elements of the domain that are required to take into account and how they may interact with this technology. Hence promote the reuse and implementation of state of art technologies in the industry.

The second area of contribution was regarding the methods used for computer surrounding understanding, that is a key element in AR applications. This research was focused on the development of training datasets, that is one of the central elements in modern computer vision (based on Machine Learning (ML)). In most of the cases, the training data used generates the major impact on the performance of ML techniques.

Nevertheless, building datasets is a complex activity that involves time and resources that are not aligned with the industrial world, that is in constant change and under considerable pressure by the market. Therefore, obtaining training datasets could be a problem especially when new products, procedures arrive constantly.

Synthetically generated datasets could be a solution to obtain almost free and fast training data, given that usually the parts and related information is already available by the companies. For instance, 3D models and visual appearance are known beforehand.

The problem with synthetically generated datasets is that usually, they underperform compared to photos based datasets because of the domain adaptation that is required to perform. Therefore in this research a method that blurs the difference between synthetic and real by simulating in a realistic manner the imperfections that usually occur in the industrial environment.

Hence, using synthetic datasets that achieve comparable performance than real data will allow reducing the burden, time and resources required to build AR applications.

In this chapter is summarized the contributions of previous chapters and discuss the future research routes. For additional information related to this topics, the reader can refer to each one of the chapters, where more detail is presented. The main contribution of this research are:

- General framework of IAR applications based on the identification of elements of the industry (Domain Variables (DV)) that could affect a technical implementation. In total, 4 Domain factors with 66 variables that influenced 5 implementation factors were identified.

  This study has been oriented to reach a general understanding of all the variables that could affect an AR implementation and to present some solutions already developed. Also, to propose to developers and researchers a global framework that could help to analyze future implementations by taking into account each one of the variables (Chapter 2).

- DV effect on surrounding understanding algorithms, in this chapter is presented the influence of the DV on technical implementations related to the processes intended to understand the surroundings.This analysis was made by first clustering the process that each one of the DV influences, and also defining what issues cause each one of them. Finally, similar issues caused by the DV (Chapter 3).

- A method for recreating relevant Domain Variables (DV) using a Physically Based Shading (PBS) approach is proposed, in order to create datasets for training and testing surrounding understanding algorithms. This method is framed under the industrial field, where the parts are very similar, present glossy effects and are subject to processes that change their visual appearance. The method allows generating fully labeled synthetic datasets specifying the distribution of the relevant variables that affect surrounding understanding algorithms (Chapter 4).

- Ablation study related to the use of the proposed method for the creation of synthetic datasets. The performed study included two types of experiments, one with unknown statistics about the target domain and other with known statistics meaning that some of the characteristics of their parts were known. Further, were found that fine-tuning models with proposed method reach comparable (No statistical difference) with models trained with photos. These results validate the proposed method as a viable alternative for training surrounding understanding algorithms applied to industrial cases (Chapter 5).

Future research can be oriented into different areas. Regarding the use synthetic datasets, the proposed method could be tested in different vision tasks such as segmentation or registration.

Similarly using the additional information produced by the synthetic dataset (e.g. pose or light direction) into the task of object recognition, it could generate more intelligent and accurate models.

Additionally, the synthetic datasets could be used to understand what is really learning the ML models. Also, help to train NN models with controlled data forcing them to learn strong features.

Further improvements to the proposed method by adding coherent backgrounds, more accurate way of setting the marginal distribution of the source dataset or simulating more accurately the extrinsic variations could lead to improvements of the performance and use to train models in more critic situations where the prediction of accuracy is more concerning.

Future work in this field could take the path of using synthetic datasets not only to increase the performance but to understand and control the features learned by the ML algorithms increasing its reliability.

Additionally implementations in another type of objects together with high-level behavioral simulations in order to be able to learn about the relationship between objects and the context. To take a next step in the understanding of the surroundings.

Further, the synthetic methods could be embedded in self-learning systems order to reinforce the learning and approaching to simulate self-teaching algorithms.

# List of Figures

# List of Tables

# Bibliography

[1] 3dwarehouse. https://3dwarehouse.sketchup.com/. Accessed: 2018-04-30.

[2] Aluminium gearknob ball. http://www.elise-shop.com/aluminium-gearknob-\ball-elise-exige-340r-all-models-w-rover-engine-p-775.html. Accessed: 2018-04-30.

[3] Bpa free crush proof plastic ball. https://www.picassotiles.com/product-page/picassotiles-kc200\-200pc-2-3inches-bpa-free-crush-proof-plastic-ball. Accessed: 2018-04-30.

[4] Gimp make seamless advanced. http://registry.gimp.org/node/28112. Accessed: 2018-04-30.

[5] Maxwell materials. http://www.maxwellrender.com/materials/. Accessed: 2018-04-30.

[6] Poliigon. www.poliigon.com/. Accessed: 2018-04-30.

[7] Textures. https://www.textures.com/. Accessed: 2018-04-30.

[8] United States. National Highway Traffic Safety Administration, T.H. Rockwell, Ohio State University. Department of Industrial, and Systems Engineering. Systems Research Group. *The Utility of Peripheral Vision to Motor Vehicle Drivers*. DOT HS-803-244. Department of Transportation, National Highway Traffic Safety Administration, 1977.

[9] Shivani Agarwal and Dan Roth. Learning a sparse representation for object detection. In *European conference on computer vision*, pages 113–127. Springer, 2002.

[10] Jamil Ahmad, Khan Muhammad, and Sung Wook Baik. Data augmentation-assisted deep learning of hand-drawn partially colored sketches for visual search. *PloS one*, 12(8):e0183838, 2017.

[11] Sabbir Ahmed. *Visual object recognition using deep convolutional neural network*. PhD thesis, BRAC University, 2017.

[12] Hugo Alvarez, Iker Aguinaga, and Diego Borro. Providing guidance for maintenance operations using automatic markerless augmented reality system. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 181–190. IEEE, 2011.

[13] Alexander Andreopoulos and John K Tsotsos. 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding*, 117(8):827–891, 2013.

[14] Juan C Arbeláez-Estrada and Gilberto Osorio-Gómez. Natural user interface for color selection in conceptual design phase. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 11(1):45–53, 2017.

[15] W Brian Arthur. Why do things become more complex. *Scientific American*, 268(5):144, 1993.

[16] J Joshan Athanesious and P Suresh. Systematic survey on object tracking methods in video. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1(8):pp–242, 2012.

[17] Mathieu Aubry and Bryan C Russell. Understanding deep features with computer-generated imagery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2875–2883, 2015.

[18] Norman R Augustine. *Augustine's laws and major system development programs*. American Institute of Aeronautics and Astronautics New York, 1983.

[19] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, 21(6):34–47, 2001.

[20] Ronald T Azuma. The most important challenge facing augmented reality. *PRESENCE: Teleoperators and Virtual Environments*, (00), 2016.

[21] Ronald T Azuma et al. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.

[22] KM Baird and Woodrow Barfield. Evaluating the effectiveness of augmented reality displays for a manual assembly task. *Virtual Reality*, 4(4):250–259, 1999.

[23] Y.B. Band. *Light and Matter: Electromagnetism, Optics, Spectroscopy and Lasers*. Light and Matter. John Wiley & Sons, 2006.

[24] Lukas Baron and Annerose Braune. Case study on applying augmented reality for process supervision in industrial use cases. In *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*, pages 1–4. IEEE, 2016.

[25] S. Beeson and J.W. Mayer. *Patterns of Light: Chasing the Spectrum from Aristotle to LEDs*. Springer New York, 2007.

[26] Amir H Behzadan, Brian W Timm, and Vineet R Kamat. General-purpose modular hardware and software framework for mobile outdoor augmented reality applications in engineering. *Advanced Engineering Informatics*, 22(1):90–105, 2008.

[27] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

[28] Thorsten Blecker and Gerhard Friedrich. *Mass customization: challenges and solutions*, volume 87. Springer Science & Business Media, 2006.

[29] Blender. Mapping types.

[30] James F Blinn. Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH Computer Graphics*, volume 11, pages 192–198. ACM, 1977.

[31] Erik Bochinski, Volker Eiselein, and Tomas Sikora. Training a convolutional neural network for multi-class object detection using solely virtual world data. In *Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on*, pages 278–285. IEEE, 2016.

[32] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[33] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *ACM SIGGRAPH*, volume 2012, pages 1–7, 2012.

[34] Pau Panareda Busto, Joerg Liebelt, and Juergen Gall. Adaptation of synthetic data for coarse-to-fine viewpoint refinement. In *BMVC*, pages 14–1, 2015.

[35] Huimin Cai, Shulong Wang, Eryun Liu, and Hongxia Liu. Invariant object recognition based on combination of sparse dbn and som with temporal trace rule. *Multimedia Tools and Applications*, 76(9):12017–12034, 2017.

[36] W.D. Callister and W.D.C. William D. *Materials Science and Engineering: An Introduction, 7th Edition Wiley Plus Set*. John Wiley & Sons, Limited, 2007.

[37] Julie Carmigniani and Borko Furht. Augmented reality: an overview. In *Handbook of augmented reality*, pages 3–46. Springer, 2011.

[38] Thomas P Caudell and David W Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on*, volume 2, pages 659–669. IEEE, 1992.

[39] Yuli Chen, Yide Ma, Dong Hwan Kim, and Sung-Kee Park. Region-based object recognition by color segmentation using a simplified pcnn. *IEEE transactions on neural networks and learning systems*, 26(8):1682–1697, 2015.

[40] Viviana Chimienti, Salvatore Iliano, Michele Dassisti, Gino Dini, and Franco Failli. Guidelines for implementing augmented reality procedures in assisting assembly operations. In *Precision Assembly Technologies and Systems*, pages 174–179. Springer, 2010.

[41] George Chryssolouris, D Mavrikios, N Papakostas, D Mourtzis, G Michalos, and K Georgoulias. Digital manufacturing: history, perspectives, and outlook. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 223(5):451–462, 2009.

[42] Andrew I Comport, Eric Marchand, Muriel Pressigout, and Francois Chaumette. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Transactions on visualization and computer graphics*, 12(4):615–628, 2006.

[43] Robert L Cook, Loren Carpenter, and Edwin Catmull. The reyes image rendering architecture. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 95–102. ACM, 1987.

[44] Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (TOG)*, 1(1):7–24, 1982.

[45] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[46] J.P. Dakin and R. Brown. *Handbook of Optoelectronics, Second Edition: Concepts, Devices, and Techniques (Volume One)*. Series in Optics and Optoelectronics. CRC Press, 2017.

[47] Dima Damen, Pished Bunnun, Andrew Calway, and Walterio W Mayol-Cuevas. Real-time learning and detection of 3d texture-less objects: A scalable approach. In *BMVC*, pages 1–12, 2012.

[48] Dima Damen, Andrew Gee, Andrew Calway, and Walterio Mayol-Cuevas. Detecting and localising multiple 3d objects: A fast and scalable approach. In *IROS Workshop on Active Semantic Perception and Object Search in the Real World (ASP-AVS-11)*, pages 1–6, 2011.

[49] Dima Damen, Andrew Gee, Walterio Mayol-Cuevas, and Andrew Calway. Egocentric real-time workspace monitoring using an rgb-d camera. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1029–1036. IEEE, 2012.

[50] E.R. Davies. *Computer Vision: Principles, Algorithms, Applications, Learning*. Elsevier Science, 2017.

[51] Fred D Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.

[52] J.R. Davis. *Corrosion: Understanding the Basics*. ASM International, 2000.

[53] Olivier L De Weck, Adam Michael Ross, and Donna H Rhodes. Investigating relationships and semantic sets amongst system lifecycle properties (ilities). 2012.

[54] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 189–198. ACM, 1998.

[55] Paul Debevec. Image-based lighting. In *ACM SIGGRAPH 2006 Courses*, page 4. ACM, 2006.

[56] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH 2008 classes*, page 32. ACM, 2008.

[57] Manjusha Deshmukh and Udhav Bhosle. A survey of image registration. *International Journal of Image Processing (IJIP)*, 5(3):245, 2011.

[58] Rashmi SG Dessai, Sufola Das Chagas Silva Araujo, Cassandra Fernandes, and PG Student. Object identification using graph theory. *International Journal of Engineering Science*, 10783, 2017.

[59] Mandar Dixit, Roland Kwitt, Marc Niethammer, and Nuno Vasconcelos. Aga: Attribute guided augmentation. *arXiv preprint arXiv:1612.02559*, 2016.

[60] Farah Mutiasari Djalal, Eef Ameel, and Gert Storms. The typicality ranking task: A new method to derive typicality judgments from children. *PLoS One*, 11(6):e0157936, 2016.

[61] Ashish Doshi, Ross T Smith, Bruce H Thomas, and Con Bouras. Use of projector based augmented reality to improve manual spot-welding precision and accuracy for automotive manufacturing. *The International Journal of Advanced Manufacturing Technology*, pages 1–15, 2016.

[62] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):692–705, 2017.

[63] Emmanuel Dubois and Laurence Nigay. Augmented reality: which augmentation for which reality? In *Proceedings of DARE 2000 on Designing augmented reality environments*, pages 165–166. ACM, 2000.

[64] Andreas Dünser and Eva Hornecker. Lessons from an ar book study. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 179–182. ACM, 2007.

[65] Charles M Eastman. *Design for X: concurrent engineering imperatives*. Springer Science & Business Media, 2012.

[66] Valerio Elia, Maria Grazia Gnoni, and Alessandra Lanzilotto. Evaluating the application of augmented reality devices in manufacturing from a process point of view: An ahp based model. *Expert Systems with Applications*, 63:187–197, 2016.

[67] Vittorio Ferrari, Tinne Tuytelaars, and Luc Van Gool. Simultaneous object recognition and segmentation by image exploration. In *European Conference on Computer Vision*, pages 40–54. Springer, 2004.

[68] Mark Fiala. Artag, a fiducial marker system using digital techniques. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 590–596. IEEE, 2005.

[69] M Fiorentino, G Monno, and AE Uva. Tangible digital master for product lifecycle management in augmented reality. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 3(2):121–129, 2009.

[70] Michele Fiorentino, Rafael Radkowski, Christian Stritzke, Antonio E Uva, and Giuseppe Monno. Design review of cad assemblies using bimanual natural interface. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 7(4):249–260, 2013.

[71] Michele Fiorentino, Antonio E Uva, Michele Gattullo, Saverio Debernardis, and Giuseppe Monno. Augmented reality on large screen for interactive maintenance instructions. *Computers in Industry*, 65(2):270–278, 2014.

[72] Michele Fiorentino, Antonio E Uva, and Giuseppe Monno. *Tangible interfaces for augmented engineering data management*. INTECH Open Access Publisher, 2010.

[73] Michele Fiorentino, Antonio E Uva, Giuseppe Monno, and Rafael Radkowski. Augmented technical drawings: a novel technique for natural interactive visualization of computer-aided design models. *Journal of Computing and Information Science in Engineering*, 12(2):024503, 2012.

[74] Pierre Fite-Georgel. Is there a reality in industrial augmented reality? In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 201–210. IEEE, 2011.

[75] A.M. Fox. *Optical Properties of Solids*. Oxford master series in condensed matter physics. Oxford University Press, 2001.

[76] Stephen J Gaukrodger and Andrew Lintott. Augmented reality and applications for assistive technology. In *Proceedings of the 1st international convention on Rehabilitation engineering & assistive technology: in conjunction with 1st Tan Tock Seng Hospital Neurorehabilitation Meeting*, pages 47–51. ACM, 2007.

[77] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[78] Seymour Giniger, Angelo Dispenzieri, and Joseph Eisenberg. Age, experience, and performance on speed and skill jobs in an applied setting. *Journal of Applied Psychology*, 68(3):469, 1983.

[79] Andrew Glassner. Building vertex normals from an unstructured polygon list. In *Graphics Gems*, pages 60–73. Elsevier, 1994.

[80] Cindy M Goral, Kenneth E Torrance, Donald P Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. In *ACM SIGGRAPH Computer Graphics*, volume 18, pages 213–222. ACM, 1984.

[81] G. Gordon. *Interior Lighting for Designers*. Interior Lighting for Designers. Wiley, 2003.

[82] Iryna Gordon and David G Lowe. What and where: 3d object recognition with accurate pose. *Toward category-level object recognition*, 4170:67–82, 2006.

[83] Dominic Gorecky, Simon F Worgan, and Gerrit Meixner. Cognito: a cognitive assistance and training system for manual tasks in industry. In *Proceedings of the 29th Annual European Conference on Cognitive Ergonomics*, pages 53–56. ACM, 2011.

[84] N.J. Gotelli and A.M. Ellison. *A Primer of Ecological Statistics*. Sinauer, 2013.

[85] Ole Gulbrandsen. Artist friendly metallic fresnel. *Journal of Computer Graphics Techniques*, 3(4), 2014.

[86] Jaewon Ha, Kyusung Cho, Francisco A Rojas, and Hyun S Yang. Real-time scalable recognition and tracking based on the server-client model for mobile augmented reality. In *VR Innovation (ISVRI), 2011 IEEE International Symposium on*, pages 267–272. IEEE, 2011.

[87] Nate Hagbi, Oriel Bergig, Jihad El-Sana, and Mark Billinghurst. Shape recognition and pose estimation for mobile augmented reality. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pages 65–71. IEEE, 2009.

[88] Mika Hakkarainen, Charles Woodward, and Mark Billinghurst. Augmented assembly using a mobile phone. In *Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on*, pages 167–168. IEEE, 2008.

[89] Michael Haller, Daniel Dobler, and Philipp Stampfl. Augmenting the reality with 3d sound sources. In *ACM SIGGRAPH 2002 conference abstracts and applications*, pages 65–65. ACM, 2002.

[90] Pengfei Han and Gang Zhao. Line-based initialization method for mobile augmented reality in aircraft assembly. *The Visual Computer*, pages 1–12, 2016.

[91] Peter A Hancock, Joseph E Mercado, James Merlo, and Jan BF Van Erp. Improving target detection in visual search through the augmenting multi-sensory cues. *Ergonomics*, 56(5):729–738, 2013.

[92] Pat Hanrahan and Wolfgang Krueger. Reflection from layered surfaces due to subsurface scattering. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 165–174. ACM, 1993.

[93] Andreas Hartl, Clemens Arth, and Dieter Schmalstieg. Instant segmentation and feature extraction for recognition of simple objects on mobile phones. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 17–24. IEEE, 2010.

[94] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[95] Bernd Heisele, Gunhee Kim, and Andrew Meyer. Object recognition with 3d models. In *BMVC*, pages 1–11, 2009.

[96] Steven Henderson and Steven Feiner. Opportunistic tangible user interfaces for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):4–16, 2010.

[97] Steven Henderson and Steven Feiner. Exploring the benefits of augmented reality documentation for maintenance and repair. *Visualization and Computer Graphics, IEEE Transactions on*, 17(10):1355–1368, 2011.

[98] Steven J Henderson and Steven Feiner. Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pages 135–144. IEEE, 2009.

[99] Steven J Henderson and Steven K Feiner. Augmented reality for maintenance and repair (armar). Technical report, DTIC Document, 2007.

[100] Steven J Henderson and Steven K Feiner. Augmented reality in the psychomotor phase of a procedural task. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 191–200. IEEE, 2011.

[101] Mauricio Hincapié, Andrea Caponio, Horacio Rios, and Eduardo González Mendívil. An introduction to augmented reality with applications in aeronautical maintenance. In *Transparent Optical Networks (ICTON), 2011 13th International Conference on*, pages 1–4. IEEE, 2011.

[102] Lei Hou and Xiangyu Wang. Experimental framework for evaluating cognitive workload of using ar system for general assembly task. In *Proceedings of the 28th International Symposium on Automation and Robotics in Construction*, 2011.

[103] Lei Hou and Xiangyu Wang. A study on the benefits of augmented reality in retaining working memory in assembly tasks: A focus on differences in gender. *Automation in Construction*, 32:38–45, 2013.

[104] Lei Hou, Xiangyu Wang, Leonhard Bernold, and Peter ED Love. Using animated augmented reality to cognitively guide assembly. *Journal of Computing in Civil Engineering*, 27(5):439–451, 2013.

[105] Lei Hou, Xiangyu Wang, and Martijn Truijens. Using augmented reality to facilitate piping assembly: an experiment-based evaluation. *Journal of Computing in Civil Engineering*, 29(1):05014007, 2013.

[106] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[107] Olivier Hugues, Philippe Fuchs, and Olivier Nannipieri. New augmented reality taxonomy: Technologies and features of augmented environment. In *Handbook of augmented reality*, pages 47–63. Springer, 2011.

[108] Wolfgang Hürst, Nina Rosa, and Jean-Paul van Bommel. Vibrotactile experiences for augmented reality. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 744–745. ACM, 2016.

[109] Pete Sawyer Ian Sommerville. *Requirements Engineering: A Good Practice Guide*. Wiley, 1997.

[110] Sony Pictures Imageworks. Physically-based shading models in film and game production.

[111] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[112] Gretchen A. Jacobson. Corrosion basics.

[113] Amin Jalali, Rammohan Mallipeddi, and Minho Lee. Sensitive deep convolutional neural network for face recognition at large standoffs with small dataset. *Expert Systems with Applications*, 2017.

[114] Stian Jensen and Andreas Løve Selvik. Using 3d graphics to train object detection systems. Master's thesis, NTNU, 2016.

[115] Seokhee Jeon and Seungmoon Choi. Haptic augmented reality: Taxonomy and an example of stiffness modulation. *Presence: Teleoperators and Virtual Environments*, 18(5):387–408, 2009.

[116] Chenfanfu Jiang, Yixin Zhu, Siyuan Qi, Siyuan Huang, Jenny Lin, Xiongwen Guo, Lap-Fai Yu, Demetri Terzopoulos, and Song-Chun Zhu. Configurable, photorealistic image rendering and ground truth synthesis by sampling stochastic grammars representing indoor scenes. *arXiv preprint arXiv:1704.00112*, 2017.

[117] Dongsik Jo and Gerard Jounghyun Kim. Ariot: scalable augmented reality framework for interacting with internet of things appliances everywhere. *IEEE Transactions on Consumer Electronics*, 62(3):334–340, 2016.

[118] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 746–753. IEEE, 2017.

[119] P.W. Jordan. *Designing Pleasurable Products: An Introduction to the New Human Factors*. Contemporary Trends Institute series. Taylor & Francis, 2002.

[120] James T Kajiya. The rendering equation. In *ACM Siggraph Computer Graphics*, volume 20, pages 143–150. ACM, 1986.

[121] Aarlenne Z Khan and J Douglas Crawford. Ocular dominance reverses as a function of horizontal gaze angle. *Vision research*, 41(14):1743–1748, 2001.

[122] JOHAN KILDAL and IÑAKI MAURTUA. Revisiting the end user's perspective in collaborative human-robot interaction. In *Advances in Cooperative Robotics: Proceedings of the 19th International Conference on Clawar 2016*, page 196. World Scientific, 2016.

[123] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[124] Gregory Kipper and Joseph Rampolla. Augmented reality, an emerging technology guide to ar, 2013.

[125] Gandjar Kiswanto and Dedy Ariansyah. Development of augmented reality (ar) for machining simulation of 3-axis cnc milling. In *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, pages 143–148. IEEE, 2013.

[126] Masashi Kitagawa and Tsuyoshi Yamamoto. 3d puzzle guidance in augmented reality environment using a 3d desk surface projection. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 133–134. IEEE, 2011.

[127] Frederick C M Kjeldsen. *Visual interpretation of hand gestures as a practical interface modality*. PhD thesis, Columbia University, 1997.

[128] Michael Kleiber and Thomas Alexander. Evaluation of a mobile ar tele-maintenance system. In *International Conference on Universal Access in Human-Computer Interaction*, pages 253–262. Springer, 2011.

[129] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[130] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[131] Joel Kronander. *Physically Based Rendering of Synthetic Objects in Real Environments*, volume 1717. Linköping University Electronic Press, 2015.

[132] Eric P Lafortune and Yves D Willems. Bi-directional path tracing. 1993.

[133] Fabrizio Lamberti, Federico Manuri, Andrea Sanna, Gianluca Paravati, Pietro Pezzolla, and Paolo Montuschi. Challenges, opportunities, and future trends of emerging techniques for augmented reality-based maintenance. *Emerging Topics in Computing, IEEE Transactions on*, 2(4):411–421, 2014.

[134] Gierad Laput, Chouchang Yang, Robert Xiao, Alanson Sample, and Chris Harrison. Em-sense: Touch recognition of uninstrumented, electrical and electromechanical objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 157–166. ACM, 2015.

[135] István Lazániy and László Szirmay-Kalos. Fresnel term approximations for metals. 2005.

[136] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[137] Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–104. IEEE, 2004.

[138] Susan J Lederman and Roberta L Klatzky. Haptic perception: A tutorial. *Attention, Perception, & Psychophysics*, 71(7):1439–1459, 2009.

[139] Hyeongmook Lee, Mark Billinghurst, and Woontack Woo. Two-handed tangible interaction techniques for composing augmented blocks. *Virtual Reality*, 15(2-3):133–146, 2011.

[140] Jung-Min Lee, Kyung-Ho Lee, Dae-Seok Kim, Gwang Lee, et al. An enhanced smart maintenance of piping system for the offshore plants. In *The Twenty-second International Offshore and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 2012.

[141] Sanghoon Lee and Ömer Akin. Augmented reality-based computational fieldwork support for equipment operations and maintenance. *Automation in Construction*, 20(4):338–352, 2011.

[142] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on statistical learning in computer vision, ECCV*, volume 2, page 7, 2004.

[143] Vincent Lepetit and Pascal Fua. Keypoint recognition using randomized trees. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1465–1479, 2006.

[144] Wing Ho Andy Li and Hongbo Fu. Augmented reflection of reality. In *ACM SIGGRAPH 2012 Emerging Technologies*, page 3. ACM, 2012.

[145] Udo Lindemann, Maik Maurer, and Thomas Braun. *Structural complexity management: an approach for the field of product design*. Springer Science & Business Media, 2008.

[146] Bo Liu, Ying Wei, Yu Zhang, and Qiang Yang. Deep neural networks for high dimension, low sample size data. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence Main track*, 2017.

[147] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[148] Filipe Costa Luz, Vasco Bila, and José Maria Dinis. Augmented reality for games. In *Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts*, pages 34–39. ACM, 2008.

[149] Shyjan Mahamud, Martial Hebert, and John Lafferty. Combining simple discriminators for object discrimination. *Computer Vision—ECCV 2002*, pages 1–22, 2002.

[150] M. Mansuripur. *Classical Optics and Its Applications*. Classical Optics and Its Applications. Cambridge University Press, 2002.

[151] S. Marschner and P. Shirley. *Fundamentals of Computer Graphics, Fourth Edition*. CRC Press, 2016.

[152] Francisco Massa, Bryan C Russell, and Mathieu Aubry. Deep exemplar 2d-3d detection by adapting from real to rendered views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6024–6033, 2016.

[153] W.R. McCluney. *Introduction to Radiometry and Photometry, Second Edition:*. Artech House applied photonics series. Artech House Publishers, 2014.

[154] Wes McDermott. *The Comprehensive PBR Guide*. Allegorithmic.

[155] Muharrem Mercimek, Kayhan Gulez, and Tarik Veli Mumcu. Real object recognition using moment invariants. *Sadhana*, 30(6):765–775, 2005.

[156] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[157] Daniel Mas Montserrat, Qian Lin, Jan Allebach, and Edward J Delp. Training object detection and recognition cnn models using data augmentation. *Electronic Imaging*, 2017(10):27–36, 2017.

[158] Roxana Moreno and Richard E Mayer. A learner-centered approach to multimedia explanations: Deriving instructional design principles from cognitive theory. *Interactive multimedia electronic journal of computer-enhanced learning*, 2(2):12–20, 2000.

[159] Michael G Morris and Viswanath Venkatesh. Age differences in technology adoption decisions: Implications for a changing work force. *Personnel psychology*, 53(2):375–403, 2000.

[160] Yair Movshovitz-Attias, Takeo Kanade, and Yaser Sheikh. How useful is photo-realistic rendering for visual learning? In *Computer Vision–ECCV 2016 Workshops*, pages 202–217. Springer, 2016.

[161] Kazuya Murakami, Ryo Kiyama, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. Poster: A wearable augmented reality system with haptic feedback and its performance in virtual assembly tasks. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 161–162. IEEE, 2013.

[162] Chikahito Nakajima and Norihiko Itho. A support system for maintenance training by augmented reality. In *Image Analysis and Processing, 2003. Proceedings. 12th International Conference on*, pages 158–163. IEEE, 2003.

[163] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[164] AYC Nee, SK Ong, G Chryssolouris, and D Mourtzis. Augmented reality applications in design and manufacturing. *CIRP Annals-Manufacturing Technology*, 61(2):657–679, 2012.

[165] Fred E Nicodemus. Directional reflectance and emissivity of an opaque surface. *Applied optics*, 4(7):767–775, 1965.

[166] Anderson Nishihara and Jun Okamoto Jr. Object recognition in assembly assisted by augmented reality system. In *SAI Intelligent Systems Conference (IntelliSys), 2015*, pages 400–407. IEEE, 2015.

[167] Jun Okamoto Jr and Anderson Nishihara. Assembly assisted by augmented reality (a3r). In *Intelligent Systems and Applications*, pages 281–300. Springer, 2016.

[168] Alex Olwal, Jonny Gustafsson, and Christoffer Lindfors. Spatial augmented reality on industrial cnc-machines. In *Electronic Imaging 2008*, pages 680409–680409. International Society for Optics and Photonics, 2008.

[169] SK Ong and ZB Wang. Augmented assembly technologies based on 3d bare-hand interaction. *CIRP Annals-Manufacturing Technology*, 60(1):1–4, 2011.

[170] SK Ong, ML Yuan, and AYC Nee. Augmented reality applications in manufacturing: a survey. *International journal of production research*, 46(10):2707–2742, 2008.

[171] Shaul Oreg. Resistance to change: developing an individual differences measure. *Journal of applied psychology*, 88(4):680, 2003.

[172] Michael Oren and Shree K Nayar. Generalization of the lambertian model and implications for machine vision. *International Journal of Computer Vision*, 14(3):227–251, 1995.

[173] G. Pahl, K. Wallace, L.T.M. Blessing, W. Beitz, and F. Bauert. *Engineering Design: A Systematic Approach*. Springer London, 2013.

[174] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[175] George Papagiannakis, Gurminder Singh, and Nadia Magnenat-Thalmann. A survey of mobile and wireless technologies for augmented reality systems. *Computer Animation and Virtual Worlds*, 19(1):3–22, 2008.

[176] Xingchao Peng and Kate Saenko. Synthetic to real adaptation with deep generative correlation alignment networks. *arXiv preprint arXiv:1701.05524*, 2017.

[177] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Learning deep object detectors from 3d models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1278–1286, 2015.

[178] Bojan Pepik, Rodrigo Benenson, Tobias Ritschel, and Bernt Schiele. What is holding back convnets for detection? In *German Conference on Pattern Recognition*, pages 517–528. Springer, 2015.

[179] Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985.

[180] Nils Petersen and Didier Stricker. Learning task structure from video examples for workflow tracking and authoring. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pages 237–246. IEEE, 2012.

[181] M. Pharr and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann series in interactive 3D technology. Morgan Kaufmann/Elsevier, 2010.

[182] Julien Pilet, Vincent Lepetit, and Pascal Fua. Real-time nonrigid surface detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 822–828. IEEE, 2005.

[183] Benjamin Planche, Ziyan Wu, Kai Ma, Shanhui Sun, Stefan Kluckner, Terrence Chen, Andreas Hutter, Sergey Zakharov, Harald Kosch, and Jan Ernst. Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5 d recognition. *arXiv preprint arXiv:1702.08558*, 2017.

[184] T. Pouli, E. Reinhard, and D.W. Cunningham. *Image Statistics in Visual Computing*. Taylor & Francis, 2013.

[185] Stuart Pugh. *Total Design: Integrated Methods for Successful Product Engineering*. Addison-Wesley, 1991.

[186] Rafael Radkowski. Investigation of visual features for augmented reality assembly assistance. In *Virtual, Augmented and Mixed Reality*, pages 488–498. Springer, 2015.

[187] Jure Ratković. *Physically based rendering*. PhD thesis, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, 2017.

[188] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer, 2016.

[189] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[190] Michelle Lenae Rusch. *Relationships between user performance and spatial ability in using map-based software on pen-based devices*. ProQuest, 2008.

[191] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[192] Juha Sääski, Tapio Salonen, Marja Liinasuo, Jarkko Pakkanen, Mikko Vanhatalo, Asko Riitahuhta, et al. Augmented reality efficiency in manufacturing industry: a case study. In *DS 50: Proceedings of NordDesign 2008 Conference, Tallinn, Estonia, 21.-23.08. 2008*, 2008.

[193] Joonas Sairiala et al. Pbr workflows in cycles render engine: Pbr workflows for realistic rendering in cycles render engine. 2015.

[194] Tapio Salonen and Juha Sääski. Dynamic and visual assembly instruction for configurable products using augmented reality techniques. In *Advanced Design and Manufacture to Gain a Competitive Edge*, pages 23–32. Springer, 2008.

[195] Andrea Sanna, Federico Manuri, Fabrizio Lamberti, Gianluca Paravati, and P Pezzolla. Using handheld devices to support augmented reality-based maintenance and assembly tasks. In *Consumer Electronics (ICCE), 2015 IEEE International Conference on*, pages 178–179. IEEE, 2015.

[196] Nicu Sebe. *Machine learning in computer vision*, volume 29. Springer Science & Business Media, 2005.

[197] J Servan, F Mas, JL Menéndez, and J Ríos. Assembly work instruction deployment using augmented reality. In *Key Engineering Materials*, volume 502, pages 25–30. Trans Tech Publ, 2012.

[198] Ali Shahrokni, Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Polyhedral object detection and pose estimation for augmented reality applications. In *Computer Animation, 2002. Proceedings of*, pages 65–69. IEEE, 2002.

[199] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[200] Huma Shoaib. A survey of augmented reality. *interactions*, 24:34.

[201] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European conference on computer vision*, pages 1–15. Springer, 2006.

[202] Kaushik Sinha et al. *Structural complexity and its implications for design of cyber-physical systems*. PhD thesis, Massachusetts Institute of Technology, 2014.

[203] Michael Smith. imagenetscraper: Bulk-download thumbnails from imagenet synsets. `https://github.com/spinda/imagenetscraper`, 2017.

[204] Ian Sommerville. *Software Engineering*. Pearson, 2005.

[205] Carsten Steger. Occlusion, clutter, and illumination invariant object recognition. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):345–350, 2002.

[206] C. Stephanidis. *User Interfaces for All: Concepts, Methods, and Tools*. Human Factors and Ergonomics. Taylor & Francis, 2000.

[207] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694, 2015.

[208] Akira Suga, Keita Fukuda, Tetsuya Takiguchi, and Yasuo Ariki. Object recognition and segmentation using sift and graph cuts. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[209] Baochen Sun and Kate Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *BMVC*, volume 1, page 3, 2014.

[210] Miao Sun, Tony X Han, Ming-Chang Liu, and Ahmad Khodayari-Rostamabad. Multiple instance learning convolutional neural networks for object recognition. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 3270–3275. IEEE, 2016.

[211] Anna Syberfeldt, Oscar Danielsson, Magnus Holm, and Lihui Wang. Visual assembling guidance using augmented reality. *Procedia Manufacturing*, 1:98–109, 2015.

[212] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[213] Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 73–80. ACM, 2003.

[214] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *arXiv preprint arXiv:1703.06907*, 2017.

[215] Amador Durán Toro. *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*. PhD thesis, Universidad de Sevilla, 2000.

[216] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011.

[217] M.A. Treiber. *An Introduction to Object Recognition: Selected Algorithms for a Wide Variety of Applications*. Advances in Computer Vision and Pattern Recognition. Springer London, 2010.

[218] Johannes Tumler, Fabian Doil, Rudiger Mecke, Georg Paul, Michael Schenk, Eberhard A Pfister, Anke Huckauf, Irina Bockelmann, and Anja Roggentin. Mobile augmented reality in industrial applications: Approaches for solution of user-related issues. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 87–90. IEEE Computer Society, 2008.

[219] Václav Tyle. Automatic recognition of texture-less objects from a single image. 2013.

[220] Emmanouil Tzorakoleftherakis, Todd D Murphey, and Robert A Scheidt. Augmenting sensorimotor control using "goal-aware" vibrotactile stimulation during reaching and manipulation behaviors. *Experimental brain research*, pages 1–12, 2016.

[221] K.T. Ulrich and S.D. Eppinger. *Product Design and Development*. McGraw-Hill /Irvin series in marketing. McGraw-Hill/Irwin, 2003.

[222] Markus Ulrich, Christian Wiedemann, and Carsten Steger. Cad-based recognition of 3d objects in monocular images. In *ICRA*, volume 9, pages 1191–1198, 2009.

[223] DWF Van Krevelen and R Poelman. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality*, 9(2):1, 2010.

[224] Eric Veach and Leonidas J Guibas. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76. ACM Press/Addison-Wesley Publishing Co., 1997.

[225] Ovidiu Vermesan, Peter Friess, Patrick Guillemin, Sergio Gusmeroli, Harald Sundmaeker, Alessandro Bassi, Ignacio Soler Jubert, Margaretha Mazura, Mark Harrison, Markus Eisenhauer, et al. Internet of things strategic research roadmap. *Internet of Things-Global Technological and Societal Trends*, 1:9–52, 2011.

[226] Nicolas Vignais, Markus Miezal, Gabriele Bleser, Katharina Mura, Dominic Gorecky, and Frédéric Marin. Innovative system for real-time ergonomic feedback in industrial manufacturing. *Applied ergonomics*, 44(4):566–574, 2013.

[227] Christian Vogel, Christoph Walter, and Norbert Elkmann. A projection-based sensor system for safe physical human-robot collaboration. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5359–5364. IEEE, 2013.

[228] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg. Real-time detection and tracking for augmented reality on mobile phones. *IEEE transactions on visualization and computer graphics*, 16(3):355–368, 2010.

[229] Raimar Wagner, Markus Thom, Roland Schweiger, Gunther Palm, and Albrecht Rothermel. Learning convolutional neural networks from few samples. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–7. IEEE, 2013.

[230] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206. Eurographics Association, 2007.

[231] Kunfeng Wang, Chao Gou, Nanning Zheng, James M Rehg, and Fei-Yue Wang. Parallel vision for perception and understanding of complex scenes: methods, framework, and perspectives. *Artificial Intelligence Review*, 48(3):299–329, 2017.

[232] X Wang, SK Ong, and AYC Nee. A comprehensive survey of augmented reality assembly research. *Advances in Manufacturing*, 4(1):1–22, 2016.

[233] Yida Wang, Can Cui, Xiuzhuang Zhou, and Weihong Deng. Zigzagnet: Efficient deep learning for real object recognition based on 3d models. In *Asian Conference on Computer Vision*, pages 456–471. Springer, 2016.

[234] Yida Wang and Weihong Deng. Self-restraint object recognition by model based cnn learning. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 654–658. IEEE, 2016.

[235] Yida Wang and Weihong Deng. Generative model with coordinate metric learning for object recognition based on 3d models. *arXiv preprint arXiv:1705.08590*, 2017.

[236] ZB Wang, LX Ng, SK Ong, and AYC Nee. Assembly planning and evaluation in an augmented reality environment. *International Journal of Production Research*, 51(23-24):7388–7404, 2013.

[237] ZB Wang, SK Ong, and AYC Nee. Augmented reality aided interactive manual assembly design. *The International Journal of Advanced Manufacturing Technology*, 69(5-8):1311–1321, 2013.

[238] Sabine Webel, Uli Bockholt, Timo Engelke, Matteo Peveri, Manuel Olbrich, and Carsten Preusche. Augmented reality training for assembly and maintenance skills. In *BIO Web of Conferences*, volume 1, page 00097. EDP Sciences, 2011.

[239] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016.

[240] Giles Westerfield, Antonija Mitrovic, and Mark Billinghurst. Intelligent augmented reality training for assembly tasks. In *International Conference on Artificial Intelligence in Education*, pages 542–551. Springer, 2013.

[241] Giles Westerfield, Antonija Mitrovic, and Mark Billinghurst. Intelligent augmented reality training for motherboard assembly. *International Journal of Artificial Intelligence in Education*, 25(1):157–172, 2015.

[242] D.E. Whitney. *Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development*. Number v. 1 in Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development. Oxford University Press, 2004.

[243] Rafal Wojciechowski, Krzysztof Walczak, Martin White, and Wojciech Cellary. Building virtual and augmented reality museum exhibitions. In *Proceedings of the ninth international conference on 3D Web technology*, pages 135–144. ACM, 2004.

[244] Lawrence B Wolff, Shree K Nayar, and Michael Oren. Improved diffuse reflection models for computer vision. *International Journal of Computer Vision*, 30(1):55–71, 1998.

[245] Yu Xiang. 3d object representations for recognition. 2016.

[246] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[247] Keiji Yanai, Ryosuke Tanno, and Koichi Okamoto. Efficient mobile implementation of a cnn-based object recognition system. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 362–366. ACM, 2016.

[248] Jincao Yao and Huimin Yu. Implicit sparse shape representation: A unified framework for object segmentation and recognition. *IEEE Signal Processing Letters*, 2012.

[249] Liu Yun and Zhang Peng. An automatic hand gesture recognition system based on viola-jones method and svms. In *Computer Science and Engineering, 2009. WCSE'09. Second International Workshop on*, volume 2, pages 72–76. IEEE, 2009.

[250] J Zhang, SK Ong, and AYC Nee. Rfid-assisted assembly guidance system in an augmented reality environment. *International Journal of Production Research*, 49(13):3919–3938, 2011.

[251] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5057–5065. IEEE, 2017.

[252] Yu-Jin Zhang. An overview of image and video segmentation in the last 40 years. *Advances in Image and Video Segmentation*, pages 1–15, 2006.

[253] Jiaping Zhao, Chin-Kai Chang, and Laurent Itti. Learning to recognize objects by retaining other factors of variation. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 560–568. IEEE, 2017.

[254] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.

[255] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 193–202. IEEE Computer Society, 2008.