STATISTICAL EVALUATION OF THE USER-SPECIFIED INPUT PARAMETERS IN AN ADAPTIVE WEIGHT DEPTH MAP ALGORITHM

STATISTICAL EVALUATION OF THE USER-SPECIFIED INPUT PARAMETERS IN AN ADAPTIVE WEIGHT DEPTH MAP ALGORITHM

Alejandro Hoyos Sierra

Supervisor:

Dr. Prof. Eng. Oscar E. Ruiz

EAFIT UNIVERSITY

COLLEGE OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

MEDELLIN

2011

STATISTICAL EVALUATION OF THE USER-SPECIFIED INPUT PARAMETERS IN AN ADAPTIVE WEIGHT DEPTH MAP ALGORITHM

Alejandro Hoyos Sierra

Graduation project submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science in Mechanical Engineering at EAFIT University

April 2011

Supervisor:
Dr. Prof. Eng. Oscar E. Ruiz

EAFIT UNIVERSITY

COLLEGE OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

MEDELLIN

2011

Acknowledgments

This graduation project was supported and guided by Dr. Prof. Eng. Oscar Ruiz from the CAD CAM CAE Laboratory where the work summarized in this document was developed.

Advisor Dr. Prof. Eng. Diego Acosta was constantly at hand to counsel on the statistical data analyses and tools, M. Sc. John Congote and M. Sc. Iñigo Barandiaran made their research results available and provided feedback to improve the presented work.

It is a pleasure to thank Sebastián Durango and the team from the CAD CAM CAE Laboratory for nourishing the research environment with their knowledge and friendship.

Finally, I am very grateful to my family for their continuous support.

Abstract

In depth map generation, the settings of the algorithm parameters to yield an accurate disparity estimation are usually chosen empirically or based on unplanned experiments. A structured statistical approach including classical and exploratory data analyses on over 14000 images to measure the relative influence of the parameters allows their tuning based on the number of bad_pixels. The implemented methodology improves the performance of dense depth map algorithms. As a result of the statistical based tuning, the algorithm improves from 16.78% to 14.48% bad_pixels rising 7 spots as per the Middlebury Stereo Evaluation Ranking Table. The performance is measured based on the distance of the algorithm results vs. the Ground Truth by Middlebury. Future work aims to achieve the tuning by using significantly smaller data sets on fractional factorial and response surface design of experiments.

Table of contents

List of Figures

List of Tables

1 Introduction

Mechanical Engineering is a broad field that includes designing or modifying devices. The engineering information of a given device can go straight from computer assisted design (CAD) blueprints to prototyping or manufacture, but the data can also be contained in physical models. Measuring and retrieving the dimensional information from the models becomes a most useful tool to bridge the development process.

Contact measurement techniques have been around for a long time, achieving a mature and robust status. Still, some situations may be better served by using non-contact measurement to preserve the model integrity or to comply with scale constraints, among other possible limitations. Laser scanning has been traditionally used for passive dimension acquisition but may be still regarded as a rather expensive alternative.

A more economical approach may be found by simply mimicking a very common passive measurement system, vision. The human eyes work like cameras taking 2D images of the same scene from very close viewpoints. The horizontal shift of a feature, when compared to the background in both images, carries part of the information required for depth estimation. The correspondence problem is solved jointly by the brain and eyes, and the estimation of depth and dimensional features is supported by geometric and visual cues.

Dimensional information based on passive measurement with cameras is presented as yet another tool to aid engineers in describing the world around them so it can be modeled and improved. Mechanical Engineers can use it to gather information for CAD, computer assisted manufacture (CAM), or computer assisted engineering (CAE).

The present work depicts a methodology to support the selection of the values of multiple parameters that can be modified by the user of a stereo reconstruction algorithm. These algorithms create new images with information of depth in an otherwise 2D image data set, and are the basis of passive measurement based on cameras.

3 Introduction

Depth map calculation deals with the estimation of multiple object depths on a given scene. It is useful for applications such as vehicle navigation, automatic surveillance, aerial cartography, passive 3D scanning, automatic industrial inspection, or 3D videoconferencing [1]. These maps are constructed by matching algorithms that generate a depth estimation at each image pixel to describe the relative distance of the surfaces on a scene.

Most depth map algorithms use two images taken with a camera shifted horizontally and parallel to its image plane between shots, keeping the optical axis orthogonal to the movement direction. The camera is calibrated following procedures like those described in [2] to remove lens distortions and produce rectified images, allowing a direct comparison of pixel rows amongst images and reducing the correspondence problem to a single dimension.

Disparity is commonly used to describe inverse depth in computer vision, and also to measure the perceived spatial shift of a feature observed from close camera viewpoints. Stereo correspondence techniques often calculate a disparity function $d(x, y)$ relating target and reference images, so that the (x, y) coordinates of the disparity space match the pixel coordinates of the reference image. Stereo methods commonly use a pair of images taken with known camera geometry to generate a dense disparity map with estimates at each pixel. This dense output is useful for applications requiring depth values even in difficult regions like occlusions and textureless areas. The ambiguity of matching pixels in heavy textured or textureless zones tends to require complex and expensive global image reasoning or statistical correlations using color and proximity measures in local support windows.

Most implementations of vision algorithms make assumptions about the visual appearance of objects in the scene to ease the matching problem. Common assumptions are that surfaces are made of smooth pieces, and features display a consistent appearance on slightly shifted camera viewpoints. The steps generally taken to compute the depth maps may include: (i) matching cost computation, (ii) cost or support aggregation, (iii) disparity computation or optimization, and (iv) disparity refinement. The algorithms begin by calculating all possible matches at all possible disparities up to a maximum that can be determined beforehand if enough information of the camera position is available from the scene. The best set of matches is then selected by optimizing a given criterion like minimizing the matching cost. Adaptive weight [3] is an implementation of a depth map

generation algorithm that matches costs over fixed moving windows.

This article is based on work done in [1] where the principles of the stereo correspondence techniques and the quantitative evaluator are discussed. The literature review is presented in section 4, followed by section 5 describing the algorithm, filters, statistical analysis and experimental set up. Results and discussions are covered in section 6, and the article is concluded in section 7.

4 Literature Review

The depth map generated by the algorithm alone can have some sparsely populated regions due to unmatched pixels or bad matches, usually requiring filters to smooth such regions during post-processing. The algorithm and filters use several user-specified parameters to generate the depth map of an image pair, and their settings are heavily influenced by the evaluated data sets [4]. Published works usually report the settings used for their specific case studies without describing the procedure followed to fine-tune them [5, 3, 6], and some explicitly state the empirical nature of these values [7]. The variation of the output as a function of several settings on selected parameters is briefly discussed while not taking into account the effect of modifying them all simultaneously [4, 3, 8]. Multiple stereo methods are compared choosing values based on experiments, but only some algorithm parameters are changed not detailing the complete rationale behind the value setting [1].

Commonly used approaches in determining the settings of depth map algorithm parameters show all or some of the following shortcomings: (i) undocumented procedures for parameter setting, (ii) lack of planning when testing for the best settings, and (iii) failure to consider interactions of changing all the parameters simultaneously.

As a response to these shortcomings, this article presents a methodology to fine-tune user-specified parameters on a depth map algorithm using a set of images from the adaptive weight implementation in [5]. Multiple settings are used and evaluated on all parameters to measure the contribution of each parameter to the output variance. A quantitative accuracy evaluation allows using main effects plots and analyses of variance on multivariate linear regression models to select the best combination of settings for each data set. The initial results are improved by setting new values of the user-specified parameters, allowing the algorithm to give much more accurate results on any rectified image pair.

5 Methodology

5.1 Image Processing

5.1.1 Depth Map Algorithm

In adaptive weight, a window is moved over each pixel on every image row, calculating a measurement based on the geometric proximity and color similarity of each pixel in the moving window to the pixel on its center. Pixels are matched on each row based on their support measurement with larger weights coming from similar pixel colors and closer pixels. The horizontal shift, or disparity, is recorded as the depth value, with higher values reflecting greater shifts and closer proximity to the camera.

The strength of grouping by color ($f_s(c_p, c_q)$) for pixels p and q is defined as the Euclidean distance between colors (Δc_{pq}) by Equation (1). Similarly, grouping strength by distance ($f_p(g_p, g_q)$) is defined as the Euclidean distance between pixel image coordinates (Δg_{pq}) by Equation (2).

$$f_s(c_p, c_q) = exp\left(-\frac{\Delta c_{pq}}{\gamma_c}\right) \tag{1}$$

where γ_c is an adjustable setting used to scale the measured color delta.

$$f_p(g_p, g_q) = exp\left(-\frac{\Delta g_{pq}}{\gamma_p}\right) \tag{2}$$

where γ_p is another adjustable parameter related to the support window size.

The matching cost between pixels shown in Equation (3) is measured by aggregating raw matching costs, using the support weights defined by Equations (1) and (2), in support windows based on both the reference and target images.

$$E(p, \bar{p}_d) = \frac{\sum_{q \in N_p, \bar{q}_d \in N_{\bar{p}_d}} w(p, q)\, w(\bar{p}_d, \bar{q}_d) \sum_{c \in \{r,g,b\}} |I_c(q) - I_c(\bar{q}_d)|}{\sum_{q \in N_p, \bar{q}_d \in N_{\bar{p}_d}} w(p, q)\, w(\bar{p}_d, \bar{q}_d)} \tag{3}$$

where $w(p,q) = f_s(c_p, c_q) \cdot f_p(g_p, g_q)$, $\bar{p}_d$ and $\bar{q}_d$ are the target image pixels at disparity d corresponding to pixels p and q in the reference image, I_c is the intensity on channels red (r), green (g), and blue (b), and N_p is the window centered at p and containing all q pixels. The size of this movable window N is another user-specified parameter. Increasing the window size reduces the chance of bad matches at the expense of missing relevant scene features.

Local methods perform most of their work on matching cost computation and aggregation, estimating the final disparity of each pixel by selecting the minimum cost value with a winner takes all optimization without any global reasoning after the dissimilarity computation.

5.1.2 Post-Processing Filters

Algorithms based on correlations depend heavily on finding similar textures at corresponding points in both reference and target images. Bad matches happen more frequently in textureless regions, occluded zones, and areas with high variation in disparity. The winner takes all approach enforces uniqueness of matches only for the reference image in such a way that points on the target image may be matched more than once, creating the need to check the disparity estimates and fill any gaps with information from neighboring pixels using post-processing filters like the ones shown in Table 1.

Filter	**Function**	**User-specified parameter**
Adaptive Weight [3]	Disparity estimation and pixel matching	γ_{aws}: similarity factor, γ_{awg}: proximity factor related to the W_{AW} pixel size of the support window
Median	Smoothing and incorrect match removal	W_M: pixel size of the median window
Cross-check[9]	Validation of disparity measurement per pixel	Δ_d: allowed disparity difference
Bilateral[10]	Intensity and proximity weighted smoothing with edge preservation	γ_{bs}: similarity factor, γ_{bg}: proximity factor related to the W_B pixel size of the bilateral window

Table 1: User-specified parameters of the adaptive weight algorithm and filters.

Median Filter. They are widely used in digital image processing to smooth signals and to remove incorrect matches and holes by assigning neighboring disparities at the

expense of edge preservation. The median filter provides a mechanism for reducing image noise, while preserving edges more effectively than a linear smoothing filter. It sorts the intensities of all the q pixels on a window of size M and selects the median value as the new intensity of the p central pixel. The size M of the window is another of the user-specified parameters.

Cross-check Filter. The correlation is performed twice by reversing the roles of the two images and considering valid only those matches having similar depth measures at corresponding points in both steps. The validity test is prone to fail in occluded areas where disparity estimates will be rejected. The allowed difference in disparities is one more adjustable parameter.

Bilateral Filter. Is a non-iterative method of smoothing images while retaining edge detail. The intensity value at each pixel in an image is replaced by a weighted average of intensity values from nearby pixels. The weighting for each pixel q is determined by the spatial distance from the center pixel p, as well as its relative difference in intensity, defined by Equation (4).

$$O_p = \frac{\sum_{q \in W} f_s\left(q - p\right) g_i\left(I_q - I_p\right) I_q}{\sum_{q \in W} f_s\left(q - p\right) g_i\left(I_q - I_p\right)} \tag{4}$$

where O is the output image, I the input image, W the weighting window, f_s the spatial weighing function, and g_i the intensity weighting function. The size of the window W is yet another parameter specified by the user.

5.2 Statistical analysis

The user-specified input parameters and output accuracy measurements data is statistically analyzed measuring the relations amongst inputs and outputs with correlation analyses, while box plots give insight on the influence of groups of settings on a given factor. A multi-variate linear regression model shown in Equation (5) relates the output variable as a function of all the parameters to find the equation coefficients, correlation of determination, and allows the analysis of variance to measure the influence of each parameter

on the output variance. Residual analyses are checked to validate the assumptions of the regression model like constant error variance, and mean of errors equal to zero [11], and if necessary, the model is transformed.

$$\hat{y} = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \epsilon \tag{5}$$

where $\hat{y}$ is the predicted variable, x_i are the factors, and β_i are the coefficients.

5.3 Experimental set up

5.3.1 Data Sets

The depth maps are calculated with an implementation developed for real time videoconferencing in [5] using well-known rectified image sets: Cones from [1], Teddy and Venus from [12], and Tsukuba head and lamp from the University of Tsukuba. The image data sets have at least a pair of rectified left and right images, and a ground truth disparity image used to compute the accuracy evaluation. Other commonly used sets are also freely available [13, 14]. The sample used consists of 14688 depth maps, 3672 for each data set. The rectified images, ground truth, and a sample of the used depth maps are shown in Figure 1.

Figure 1: Image Data Sets. Top to bottom: Cones, Teddy, Tsukuba, Venus. (a) Rectified left image, (b) Ground truth, (c) No filters depth map, (d) All low settings depth map, and (e) All high settings depth map.

The user-specified parameters have a working range where they can be set so that the input changes are detected on the output variation. Input factors of the selected adaptive

weight algorithm, its post-processing filters, and the data sets are shown in Table 2. Coding the levels of each factor from -1 at its minimum value to $+1$ at its maximum, and proportionally scaling any other values in between, allows the direct comparison of the contribution of each factor to the output variance by checking at the regression model coefficients.

Parameter	**Name**	**Levels**	**Values**	**Coding**
Adaptive Weights Window Size	*aw_win*	4	[1 3 5 7]	[-1 -0.3 0.3 1]
Adaptive Weights Color Factor	*aw_col*	6	[4 7 10 13 16 19]	[-1 -0.6 -0.2 0.2 0.6 1]
Median Window Size	*m_win*	3	[*N/A* 3 5]	[*N/A* -1 0.2 1]
Cross-Check Disparity Delta	*cc_disp*	4	[*N/A* 0 1 2]	[*N/A* -1 0 1]
Cross-Bilateral Window Size	*cb_win*	5	[*N/A* 1 3 5 7]	[*N/A* -1 -0.3 0.3 1]
Cross-Bilateral Color Factor	*cb_col*	7	[*N/A* 4 7 10 13 16 19]	[*N/A* -1 -0.6 -0.2 0.2 0.6 1]

Table 2: User-specified Parameter Tested Settings and Coding

5.3.2 Evaluation

Many recent stereo correspondence performance studies use the Middlebury Stereomatcher for their quantitative comparisons [4, 8, 15]. The evaluator code, sample scripts, and image data sets are available from the Middlebury stereo vision site [16], providing a flexible and standard platform for easy evaluation. The local set-up, testing, and validation of the evaluator helps ensure the reproducibility of the results.

The Middlebury evaluator quantifies the root mean square errors (RMS) and proportion of bad pixels over typical problem regions as presented in Table 3. The quality of the computed correspondences is quantified with a performance evaluation based on the known ground truth data. The percentage of bad pixels is preferred over RMS disparity errors since it gives a better indication of the overall performance, for an algorithm may have low bad pixels and large RMS error due to the poor matches on just a few regions; meaning RMS errors are more relevant at very low bad pixel percentages [1]. The online Middlebury stereo evaluation table gives a visual indication of how well the methods perform with the proportion of bad pixels (*bad_pixels*) metric defined as the average of the proportion of bad pixels in the whole image (*bad_pixels_all*), the proportion of bad pixels in non-occluded regions (*bad_pixels_nonocc*), and the proportion of bad pixels in areas near depth discontinuities (*bad_pixels_discont*) in all data sets.

Parameter	**Description**
rms_error_all	Root Mean Square (RMS) disparity error (all pixels)
rms_error_nonocc	RMS disparity error (non-occluded pixels only)
rms_error_occ	RMS disparity error (occluded pixels only)
rms_error_textured	RMS disparity error (textured pixels only)
rms_error_textureless	RMS disparity error (textureless pixels only)
rms_error_discont	RMS disparity error (near depth discontinuities)
bad_pixels_all	Fraction of bad points (all pixels)
bad_pixels_nonocc	Fraction of bad points (non-occluded pixels only)
bad_pixels_occ	Fraction of bad points (occluded pixels only)
bad_pixels_textured	Fraction of bad points (textured pixels only)
bad_pixels_textureless	Fraction of bad points (textureless pixels only)
bad_pixels_discont	Fraction of bad points (near depth discontinuities)
evaluate_only	Read specified depth map and evaluate only
output_params	Text file logging all used parameters
depth_map	Evaluated image

Table 3: Result metrics computed by the Middlebury Stereomatcher evaluator.

6 Results and Discussion

6.1 Classical Data Analysis

6.1.1 Variable Selection

Pearson correlation of the factors show that they are independent and that each one must be included in the evaluation. On the other hand, a strong correlation amongst *bad_pixels* and the other outputs is detected and shown in Figure 2. This allows the selection of *bad_pixels* as the sole output because the other responses are expected to follow a similar trend.

6.1.2 Linear Model

The analysis of variance on a multi-variate linear regression (MVLR) over all data sets using the most parsimonious model quantifies the parameters with the most influence as shown in Figure 3. *cc_disp* is the most significant factor accounting for a third to a half of the variance on every case.

Interactions and higher order terms are included on the multi-variate linear regression models to improve the goodness of fit as shown in Table 4. Reducing the number of input images per dataset from 3456 to 1526 by excluding the worst performing cases corresponding to $cc\_disp = 0$ and $aw\_col = [4, 7]$, allows the use of a cubic model with interactions having 13 terms and an almost perfect fit of 99.05% after testing for the best nested model.

$$\begin{aligned} bad\_pixels = & -0.0205 aw\_win^3 - 0.001 cc\_disp^3 + 0.0008 cb\_win^3 + 0.0417 aw\_win^2 \\ & + 0.0176 aw\_col^2 + 0.0043 cb\_win^2 - 0.0654 aw\_win \\ & - 0.0481 aw\_col - 0.0105 m\_win - 0.0067 aw\_win \cdot aw\_col \\ & + 0.0076 aw\_win \cdot m\_win + 0.0023 aw\_col \cdot m\_win + 0.259 \end{aligned} \tag{6}$$

The residuals of the selected model fail to follow a normal distribution as shown in Figure 4. Transforming the output variable or removing large residuals does not improve the residuals distribution, and there are no reasons to exclude any outliers from the database

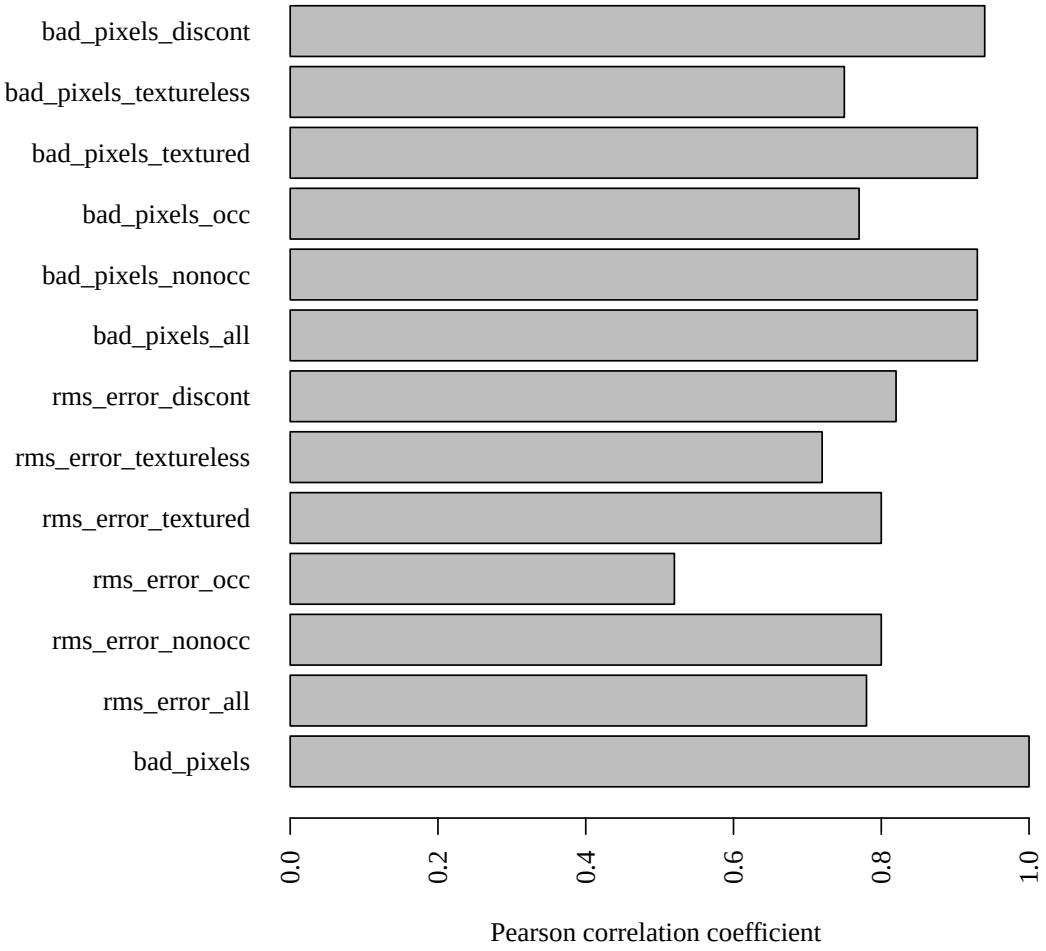


Figure 2: Output Correlations to *bad_pixels*

as outliers. Nonetheless, improved algorithm performance settings are found using the model to obtain lower *bad_pixels* values.

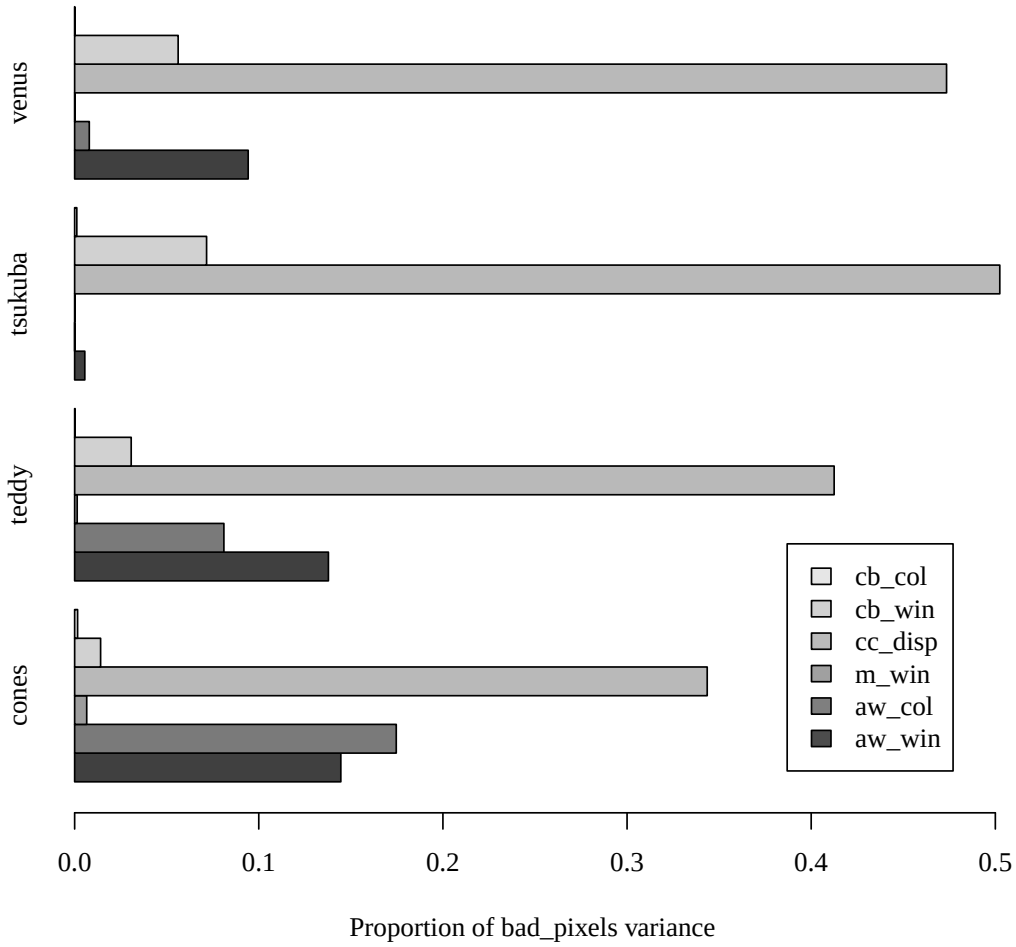


Figure 3: Parameter contribution to the *bad_pixels* variance

6.2 Exploratory Data Analysis

6.2.1 Box Plots

Box plots analysis of *bad_pixels* presented in Figure 5 show lower output values from using filters, relaxed cross-check disparity delta values, large adaptive weight window sizes, and

Images	Model	Order	Int.	Terms	R^2
3456	Linear+1	1	1	6	63.31%
	Quadratic+1	2	1	10	81.34%
	Cubic+1	3	1	12	81.49%
	Linear+2	1	2	16	73.22%
	Quadratic+2	2	2	20	91.26%
	Cubic+2	3	2	22	91.41%
	Linear+3	1	3	22	74.75%
	Quadratic+3	2	3	26	92.78%
	Cubic+3	3	3	28	92.93%
1536	Linear+1	1	1	4	90.81%
	Quadratic+1	2	1	7	97.94%
	Cubic+1	3	1	10	98.29%
	Linear+2	1	2	7	91.58%
	Quadratic+2	2	2	10	98.71%
	Cubic+2	3	2	13	99.05%

Table 4: Multi-variate linear regression nested model comparison.

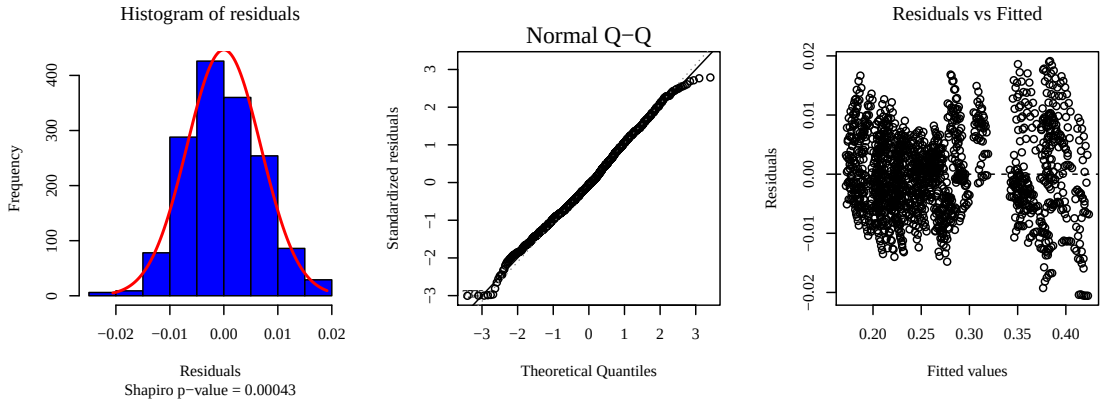


Figure 4: Residual plots

large adaptive weight color factor values. The median window size, bilateral window size, and bilateral window color values do not show a significant influence on the output at the studied levels.

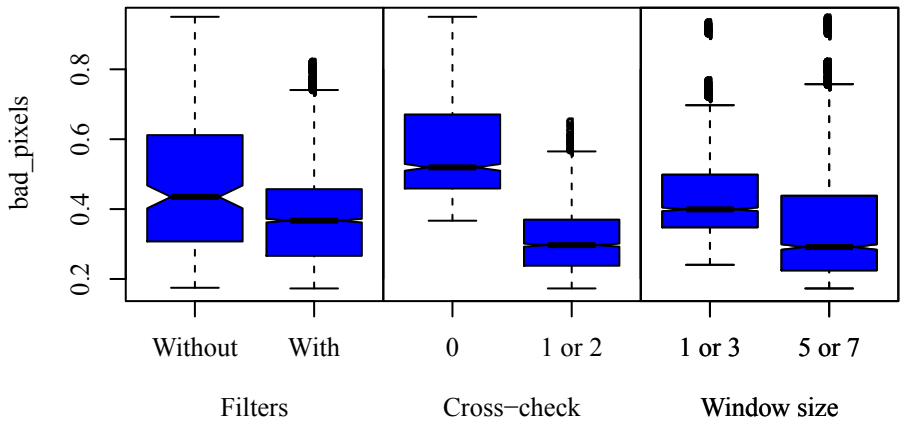


Figure 5: Box plots showing differences on the *bad_pixels* output measure from grouped factor levels.

6.2.2 Main Effects Plots

The influence of the parameters is also shown on the slopes of the main effects plots of Figure 6 and confirms the behavior found with the ANOVA of the multi-variate linear regression model. The settings to lower *bad_pixels* from this analysis yields a result of 14.48%.

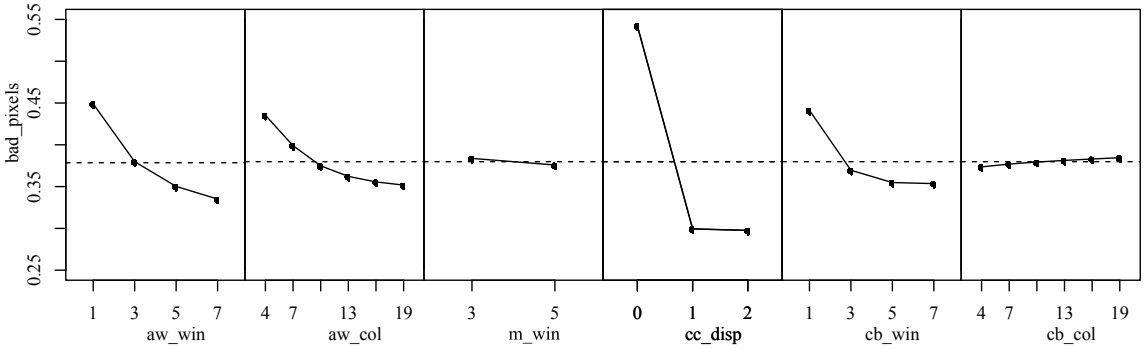


Figure 6: Main Effects Plots of each factor level for all data sets. Steeper slopes relate to bigger influence on the variance of the *bad_pixels* output measurement.

In summary, the most noticeable influence on the output variable comes from having a relaxed cross-check filter, accounting for nearly half the response variance in all the study data sets. Larger window sizes comes as the next most influential factor, followed by larger color factor, and finally bigger window sizes on the bilateral filter. Increasing the window sizes on the main algorithm yield better overall results at the expense of longer

running times and some foreground loss of sharpness, the support weights on each pixel have the chance of becoming more distinct and potentially reduce disparity mismatches. Increasing the color factor on the main algorithm allows better results by reducing the color differences, and slightly compensating minor variations in intensity from different viewpoints.

A small median smoothing filter window size is faster than a bigger one, while still having a similar accuracy. Low settings on both the window size and the color factor on the bilateral filter seem to work best for a good balance between performance and accuracy.

The optimal settings in the original data set are presented in Table 5 along with the proposed combinations:

Low settings comprise the depth maps with all their parameter settings at each of their minimum tested values yielding 67.62% *bad_pixels*.

High settings relates to depth maps with all their parameter settings at each of their maximum tested values yielding 19.84% *bad_pixels*.

Best initial are the most accurate depth maps from the study data set yielding 16.78% *bad_pixels*.

Exploratory analysis corresponds to the settings determined using the exploratory data analysis based on box plots and main effects plots yielding 14.48% *bad_pixels*.

MVLR optimization is the extrapolation optimization of the classical data analysis based on multi-variate linear regression model, nested models, and ANOVA yielding 14.66% *bad_pixels*.

The exploratory analysis estimation and the MVLR optimization tend to converge at similar lower *bad_pixels* values using the same image data set. The best initial and improved depth map outputs are shown in Figure 7.

Data set	Run Type	bad_pixels	aw_win	aw_col	m_win	cc_disp	cb_win	cb_col
Cones	Low Settings	69.60%	1	4	3	0	1	4
	High Settings	24.69%	7	19	5	2	7	19
	Best Initial	20.76%	7	19	5	1	3	4
	Quad. Interpolation	20.71%	7	19	5	2	2	4
	Quad. Extrapolation	17.09%	11	22	5	3	3	18
	Main Effect Estimation	16.92%	9	22	5	1	3	4
Teddy	Low Settings	73.77%	1	4	3	0	1	4
	High Settings	25.56%	7	19	5	2	7	19
	Best initial	22.81%	7	19	5	1	3	4
	Quad. Interpolation	22.88%	7	19	5	2	2	4
	Quad. Extrapolation	20.18%	11	22	5	3	3	18
	Main Effect Estimation	21.25%	9	22	5	1	5	4
Tsukuba	Low Settings	59.71%	1	4	3	0	1	4
	High Settings	14.21%	7	19	5	2	7	19
	Best initial	9.68%	7	16	5	1	3	4
	Quad. Interpolation	9.62%	7	19	5	2	2	4
	Quad. Extrapolation	9.93%	11	22	5	3	3	18
	Main Effect Estimation	9.35%	9	22	5	1	5	4
Venus	Low Settings	67.39%	1	4	3	0	1	4
	High Settings	14.89%	7	19	5	2	7	19
	Best initial	13.20%	7	16	5	1	7	4
	Quad. Interpolation	14.14%	7	19	5	2	2	4
	Quad. Extrapolation	11.42%	11	22	5	3	3	18
	Main Effect Estimation	10.37%	9	16	5	1	9	4
All	Low Settings	67.62%	1	4	3	0	1	4
	High Settings	19.84%	7	19	5	2	7	19
	Best Initial	16.78%	7	19	5	1	3	4
	Quad. Interpolation	16.84%	7	19	5	2	2	4
	Quad. Extrapolation	14.66%	11	22	5	3	3	18
	Main Effect Estimation	14.48%	9	22	5	1	3	4

Table 5: Best initial and proposed settings comparison. *bad_pixels* values and factor level settings for the best combination in the initial data set and the proposed combination based on the factor influence analysis.

Figure 7: Depth Map Comparison. Top: best initial, bottom: new settings. (a) Cones, (b) Teddy, (c) Tsukuba, and (d) Venus data set.

2 Statistical Evaluation of the User-Specified Input Parameters in an Adaptive Weight Depth Map Algorithm

Alejandro Hoyos[a], John Congote[a b], Iñigo Barandiaran[b],
Diego Acosta[c], Oscar Ruiz[a]

[a] *CAD CAM CAE Laboratory, EAFIT University, Medellin, Colombia*

[b] *Vicomtech Research Center, Donostia-San Sebastián, Spain*

[c] *DDP Research Group, EAFIT University, Medellin, Colombia*

2.1 Context

The CAD CAM CAE Laboratory at EAFIT University works in close relation with other research institutes like Vicomtech in Spain. One of the Laboratory members is pursuing his doctoral degree at EAFIT while taking part in projects developed at Vicomtech like enhanced tele-presence. This research includes real time 3D visualization implemented with a variant of adaptive weight stereo correspondence algorithm using median smoothing, disparity cross-check, and bilateral filtering. The real time depth map generation architecture was used to create a set of depth maps with multiple combinations of input factor levels.

Contents of this graduation project are included in the article "Statistical tuning of Depth Map Algorithms" by Alejandro Hoyos, John Congote, Iñigo Barandiaran, Diego Acosta, and Oscar Ruiz; submitted to the 14^{th} International Conference on Computer Analysis of Images and Patterns to be held in Seville, Spain on August 29 to 31 of 2011. As co-authors of such publication, we give our permission for this material to appear in this graduation project. We are ready to provide any additional information on the subject, as needed.

Prof. Dr. Eng. Oscar E. Ruiz
oruiz@eafit.edu.co
Coordinator CAD CAM CAE Laboratory
EAFIT University, Medellin, Colombia

Prof. Dr. Eng. Diego A. Acosta
dacostam@eafit.edu.co
DDP Research Group
EAFIT University, Medellin, Colombia

M. Sc. John E. Congote
jcongote@eafit.edu.co
Vicomtech Research Center
Ph. D. student at
EAFIT University, Medellin, Colombia

M. Sc. Iñigo Barandiaran
ibarandiaran@vicomtech.org
Vicomtech Research Center

7 Conclusions

This work used an existing database of depth map images with multiple input parameter values in the adaptive weights algorithm, and the median, cross-check and bilateral post-processing filters. The classical statistical approach using analyses of variance on multivariate linear regression models complemented with box plots and main effects plots from the exploratory data analysis is presented as a structured methodology to measure the relative influence of each factor on the output variance and the identification of new settings to improve the results from 16.78% to 14.48% bad_pixels. The methodology is applicable on any group of depth map image sets generated with an algorithm where the relative influence of the user-specified parameters merits to be assessed.

8 Future Work

Using design of experiments to drive the generation of depth maps reduces the number of depth maps needed to carry out the study when a large previously generated database is not available or practical to build. Further analysis on the input factors should be started with exploratory experimental factorial designs comprising the full range on each factor, followed by a response surface experimental design and analysis. In selecting the factor levels, analyzing the influence of each filter independently would be an interesting criterion.

9 Acknowledgments

This work has been partially supported by the Spanish Administration Agency CDTI under project CENIT-VISION 2007-1007, the Colombian Administrative Department of Science, Technology, and Innovation; and the Colombian National Learning Service (COLCIENCIAS-SENA) grant No. 1216-479-22001.

References

[1] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, 2002.

[2] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. In Lawrence B. Wolff, Steven A. Shafer, and Glenn Healey, editors, *Radiometry*, pages 221–244. Jones and Bartlett Publishers, Inc., , USA, 1992.

[3] Kuk-Jin Yoon and In So Kweon. Adaptive support-weight approach for correspondence search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):650, 2006.

[4] Minglun Gong, Ruigang Yang, Liang Wang, and Mingwei Gong. A performance study on different cost aggregation approaches used in real-time stereo matching. *Int. J. Comput. Vision*, 75:283–296, November 2007.

[5] John Congote, Iñigo Barandiaran, Javier Barandiaran, Tomas Montserrat, Julien Quelen, Christian Ferrán, Pere J. Mindan, Olga Mur, Francesc Tarrés, and O. Ruiz. Real-time depth map generation architecture for 3d videoconferencing. In *3DTV-Conference: The True Vision - Capture, Transmissionand Display of 3D Video (3DTV-CON), 2010*, pages 1–4, Tampere, Finland, June 2010.

[6] Zheng Gu, Xianyu Su, Yuankun Liu, and Qican Zhang. Local stereo matching with adaptive support-weight, rank transform and disparity calibration. *Pattern Recogn. Lett.*, 29:1230–1235, July 2008.

[7] Asmaa Hosni, Michael Bleyer, Margrit Gelautz, and Christoph Rhemann. Local stereo matching using geodesic support weights. In *Proceedings of the 16th IEEE Int. Conf. on Image Processing (ICIP)*, pages 2093–2096, New York, NY, USA, November 2009. Elsevier Science Inc.

[8] Liang Wang, Mingwei Gong, Minglun Gong, and Ruigang Yang. How far can we go with local optimization in real-time stereo matching. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 3DPVT '06, pages 129–136, Washington, DC, USA, 2006. IEEE Computer Society.

[9] Pascal Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1):35–49, December 1993.

[10] Ben Weiss. Fast median and bilateral filtering. *ACM Trans. Graph.*, 25:519–526, July 2006.

[11] NIST/SEMATECH. e-handbook of statistical methods. `http://www.itl.nist.gov/div898/handbook/`, 2010.

[12] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:195–202, June 2003.

[13] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007.

[14] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007.

[15] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[16] Daniel Scharstein and Richard Szeliski. Middlebury stereo vision page. `http://vision.middlebury.edu/stereo/`, 2007.

10 Appendix A

Article by Alejandro Hoyos, John Congote, Iñigo Barandiaran, Diego Acosta, and Oscar Ruiz submitted to the 14^{th} International Conference on Computer Analysis of Images and Patterns to be held in Seville, Spain on August 29 to 31 of 2011.

Statistical tuning of Depth Map Algorithms

Alejandro Hoyos[1], John Congote[1,2], Iñigo Barandiaran[2], Diego Acosta[3], and Oscar Ruiz[1]

[1] CAD CAM CAE Laboratory, EAFIT University, Medellin, Colombia
{ahoyossi, oruiz}@eafit.edu.co,
[2] Vicomtech Research Center, Donostia-San Sebastián, Spain
{jcongote, ibarandiaran}@vicomtech.org,
[3] DDP Research Group, EAFIT University, Medellin, Colombia
dacostam@eafit.edu.co

Abstract. In depth map generation, the settings of the algorithm parameters to yield an accurate disparity estimation are usually chosen empirically or based on unplanned experiments. A structured statistical approach including classical and exploratory data analyses on over 14000 images to measure the relative influence of the parameters allows their tuning based on the number of *bad_pixels*. The implemented methodology improves the performance of dense depth map algorithms. As a result of the statistical based tuning, the algorithm improves from 16.78% to 14.48% *bad_pixels* rising 7 spots as per the Middlebury Stereo Evaluation Ranking Table. The performance is measured based on the distance of the algorithm results vs. the Ground Truth by Middlebury. Future work aims to achieve the tuning by using significantly smaller data sets on fractional factorial and response surface design of experiments.

Keywords: Stereo Image Processing; Parameter Estimation; Depth Map

1 Introduction

Depth map calculation deals with the estimation of multiple object depths on a scene. It is useful for applications like vehicle navigation, automatic surveillance, aerial cartography, passive 3D scanning, automatic industrial inspection, or 3D videoconferencing [1]. These maps are constructed by generating, at each pixel, an estimation of the distance between the screen and the object surface (depth).

Disparity is commonly used to describe inverse depth in computer vision, and also to measure the perceived spatial shift of a feature observed from close camera viewpoints. Stereo correspondence techniques often calculate a disparity function $d(x, y)$ relating target and reference images, so that the (x, y) coordinates of the disparity space match the pixel coordinates of the reference image. Stereo methods commonly use a pair of images taken with known camera geometry to generate a dense disparity map with estimates at each pixel. This dense output is useful for applications requiring depth values even in difficult regions like occlusions and textureless areas. The ambiguity of matching pixels in heavy textured or textureless zones tends to require complex and expensive global

image reasoning or statistical correlations using color and proximity measures in local support windows.

Most implementations of vision algorithms make assumptions about the visual appearance of objects in the scene to ease the matching problem. The steps generally taken to compute the depth maps may include: (i) matching cost computation, (ii) cost or support aggregation, (iii) disparity computation or optimization, and (iv) disparity refinement.

This article is based on work done in [1] where the principles of the stereo correspondence techniques and the quantitative evaluator are discussed. The literature review is presented in section 2, followed by section 3 describing the algorithm, filters, statistical analysis and experimental set up. Results and discussions are covered in section 4, and the article is concluded in section 5.

2 Literature Review

The algorithm and filters use several user-specified parameters to generate the depth map of an image pair, and their settings are heavily influenced by the evaluated data sets [2]. Published works usually report the settings used for their specific case studies without describing the procedure followed to fine-tune them [3–5], and some explicitly state the empirical nature of these values [6]. The variation of the output as a function of several settings on selected parameters is briefly discussed while not taking into account the effect of modifying them all simultaneously [3, 2, 7]. Multiple stereo methods are compared choosing values based on experiments, but only some algorithm parameters are changed not detailing the complete rationale behind the value setting [1].

Conclusions of the Literature Review. Commonly used approaches in determining the settings of depth map algorithm parameters show all or some of the following shortcomings: (i) undocumented procedures for parameter setting, (ii) lack of planning when testing for the best settings, and (iii) failure to consider interactions of changing all the parameters simultaneously.

As a response to these shortcomings, this article presents a methodology to fine-tune user-specified parameters on a depth map algorithm using a set of images from the adaptive weight implementation in [4]. Multiple settings are used and evaluated on all parameters to measure the contribution of each parameter to the output variance. A quantitative accuracy evaluation allows using main effects plots and analyses of variance on multi-variate linear regression models to select the best combination of settings for each data set. The initial results are improved by setting new values of the user-specified parameters, allowing the algorithm to give much more accurate results on any rectified image pair.

3 Methodology

Image Processing. In the adaptive weight algorithm, a window is moved over each pixel on every image row, calculating a measurement based on the geometric proximity and color similarity of each pixel in the moving window to the pixel

on its center. Pixels are matched on each row based on their support measurement with larger weights coming from similar pixel colors and closer pixels. The horizontal shift, or disparity, is recorded as the depth value, with higher values reflecting greater shifts and closer proximity to the camera.

The strength of grouping by color ($f_s(c_p, c_q)$) for pixels p and q is defined as the Euclidean distance between colors (Δc_{pq}) by Equation (1). Similarly, grouping strength by distance ($f_p(g_p, g_q)$) is defined as the Euclidean distance between pixel image coordinates (Δg_{pq}) by Equation (2). Where γ_c and γ_p are adjustable settings used to scale the measured color delta and window size respectively.

$$f_s(c_p, c_q) = exp\left(-\frac{\Delta c_{pq}}{\gamma_c}\right) \tag{1}$$

$$f_p(g_p, g_q) = exp\left(-\frac{\Delta g_{pq}}{\gamma_p}\right) \tag{2}$$

The matching cost between pixels shown in Equation (3) is measured by aggregating raw matching costs, using the support weights defined by Equations (1) and (2), in support windows based on both the reference and target images.

$$E(p, \bar{p}_d) = \frac{\sum_{q \in N_p, \bar{q}_d \in N_{\bar{p}_d}} w(p,q)\, w(\bar{p}_d, \bar{q}_d) \sum_{c \in \{r,g,b\}} |I_c(q) - I_c(\bar{q}_d)|}{\sum_{q \in N_p, \bar{q}_d \in N_{\bar{p}_d}} w(p,q)\, w(\bar{p}_d, \bar{q}_d)} \tag{3}$$

where $w(p,q) = f_s(c_p, c_q) \cdot f_p(g_p, g_q)$, $\bar{p}_d$ and $\bar{q}_d$ are the target image pixels at disparity d corresponding to pixels p and q in the reference image, I_c is the intensity on channels red (r), green (g), and blue (b), and N_p is the window centered at p and containing all q pixels. The size of this movable window N is another user-specified parameter. Increasing the window size reduces the chance of bad matches at the expense of missing relevant scene features.

Algorithms based on correlations depend heavily on finding similar textures at corresponding points in both reference and target images. Bad matches happen more frequently in textureless regions, occluded zones, and areas with high variation in disparity. The winner takes all approach enforces uniqueness of matches only for the reference image in such a way that points on the target image may be matched more than once, creating the need to check the disparity estimates and fill any gaps with information from neighboring pixels using post-processing filters like the ones shown in Table 1.

Statistical analysis. The user-specified input parameters and output accuracy measurements data is statistically analyzed measuring the relations amongst inputs and outputs with correlation analyses, while box plots give insight on the influence of groups of settings on a given factor. A multi-variate linear regression model shown in Equation (4) relates the output variable as a function of all the parameters to find the equation coefficients, correlation of determination, and allows the analysis of variance to measure the influence of each parameter on the output variance. Residual analyses are checked to validate the assumptions

Filter	Function	User-specified parameter
Adaptive Weight [3]	Disparity estimation and pixel matching	γ_{aws}: similarity factor, γ_{awg}: proximity factor related to the W_{AW} pixel size of the support window
Median	Smoothing and incorrect match removal	W_M: pixel size of the median window
Cross-check[8]	Validation of disparity measurement per pixel	Δ_d: allowed disparity difference
Bilateral[9]	Intensity and proximity weighted smoothing with edge preservation	γ_{bs}: similarity factor, γ_{bg}: proximity factor related to the W_B pixel size of the bilateral window

Table 1. User-specified parameters of the adaptive weight algorithm and filters.

of the regression model like constant error variance, and mean of errors equal to zero, and if necessary, the model is transformed. The parameters are normalized to fit the range $(-1, 1)$ as show in Table 2.

$$\hat{y} = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \epsilon \tag{4}$$

where $\hat{y}$ is the predicted variable, x_i are the factors, and β_i are the coefficients.

Experimental set up. The depth maps are calculated with an implementation developed for real time videoconferencing in [4]. Using well-known rectified image sets: Cones from [1], Teddy and Venus from [10], and Tsukuba head and lamp from the University of Tsukuba. Other commonly used sets are also freely available [11, 12]. The sample used consists of 14688 depth maps, 3672 for each data set, like the ones shown in Figure 1.

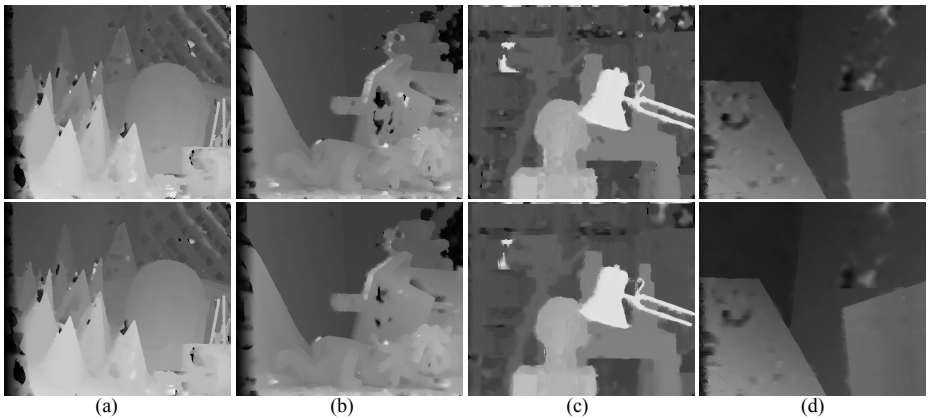


Fig. 1. Depth Map Comparison. Top: best initial, bottom: new settings. (a) Cones, (b) Teddy, (c) Tsukuba, and (d) Venus data set.

Parameter	Name	Levels	Values	Coding
Adaptive Weights Window Size	*aw_win*	4	[1 3 5 7]	[-1 -0.3 0.3 1]
Adaptive Weights Color Factor	*aw_col*	6	[4 7 10 13 16 19]	[-1 -0.6 -0.2 0.2 0.6 1]
Median Window Size	*m_win*	3	[*N/A* 3 5]	[*N/A* -1 0.2 1]
Cross-Check Disparity Delta	*cc_disp*	4	[*N/A* 0 1 2]	[*N/A* -1 0 1]
Cross-Bilateral Window Size	*cb_win*	5	[*N/A* 1 3 5 7]	[*N/A* -1 -0.3 0.3 1]
Cross-Bilateral Color Factor	*cb_col*	7	[*N/A* 4 7 10 13 16 19]	[*N/A* -1 -0.6 -0.2 0.2 0.6 1]

Table 2. User-specified parameters of the adaptive weight algorithm.

Many recent stereo correspondence performance studies use the Middlebury Stereomatcher for their quantitative comparisons [2, 7, 13]. The evaluator code, sample scripts, and image data sets are available from the Middlebury stereo vision site[4], providing a flexible and standard platform for easy evaluation. The online Middlebury Stereo Evaluation Table gives a visual indication of how well the methods perform with the proportion of bad pixels (*bad_pixels*) metric defined as the average of the proportion of bad pixels in the whole image (*bad_pixels_all*), the proportion of bad pixels in non-occluded regions (*bad_pixels_nonocc*), and the proportion of bad pixels in areas near depth discontinuities (*bad_pixels_discont*) in all data sets.

4 Results and Discussion

Variable selection. Pearson correlation of the factors show that they are independent and that each one must be included in the evaluation. On the other hand, a strong correlation amongst *bad_pixels* and the other outputs is detected and shown in Figure 2(b). This allows the selection of *bad_pixels* as the sole output because the other responses are expected to follow a similar trend.

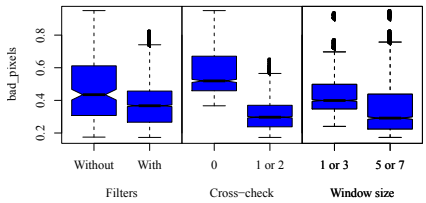
Exploratory Data Analysis. Box plots analysis of *bad_pixels* presented in Figure 2(a) show lower output values from using filters, relaxed cross-check disparity delta values, large adaptive weight window sizes, and large adaptive weight color factor values. The median window size, bilateral window size, and bilateral window color values do not show a significant influence on the output at the studied levels.

The influence of the parameters is also shown on the slopes of the main effects plots of Figure 3 and confirms the behavior found with the ANOVA of the multi-variate linear regression model. The settings to lower *bad_pixels* from this analysis yields a result of 14.48%.

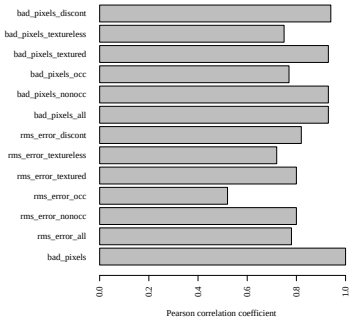
Multi-variate linear regression model. The analysis of variance on a multi-variate linear regression (MVLR) over all data sets using the most parsimonious model quantifies the parameters with the most influence as shown in Figure 2(c). *cc_disp* is the most significant factor accounting for a third to a half of the variance on every case.

Interactions and higher order terms are included on the multi-variate linear regression models to improve the goodness of fit. Reducing the number of input

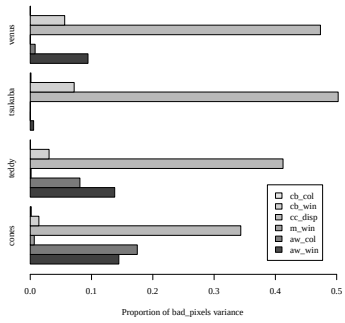
[4] http://vision.middlebury.edu/stereo/



(a) Box Plots



(b) Correlation



(c) ANOVA

Fig. 2. 2(a) Box Plots of *bad_pixels*, 2(b) *bad_pixels* and other outputs correlation, and 2(b) Contribution to the *bad_pixels* variance by parameter.

images per dataset from 3456 to 1526 by excluding the worst performing cases corresponding to $cc\_disp = 0$ and $aw\_col = [4, 7]$, allows using a cubic model with interactions and an R^2 of 99.05%.

The residuals of the selected model fail to follow a normal distribution. Transforming the output variable or removing large residuals does not improve the residuals distribution, and there are no reasons to exclude any outliers from the image data set. Nonetheless, improved algorithm performance settings are found using the model to obtain lower *bad_pixels* values comparable to the ones obtained through the exploratory data analysis (14.66% vs. 14.48%).

In summary, the most noticeable influence on the output variable comes from having a relaxed cross-check filter, accounting for nearly half the response variance in all the study data sets. Larger window sizes comes as the next most influential factor, followed by larger color factor, and finally bigger window sizes on the bilateral filter. Increasing the window sizes on the main algorithm yield better overall results at the expense of longer running times and some foreground loss of sharpness, while the support weights on each pixel have the chance of becoming more distinct and potentially reduce disparity mismatches. Increasing the color factor on the main algorithm allows better results by reducing the color differences, and slightly compensating minor variations in intensity from different viewpoints.

A small median smoothing filter window size is faster than a larger one, while still having a similar accuracy. Low settings on both the window size and the color factor on the bilateral filter seem to work best for a good balance between performance and accuracy.

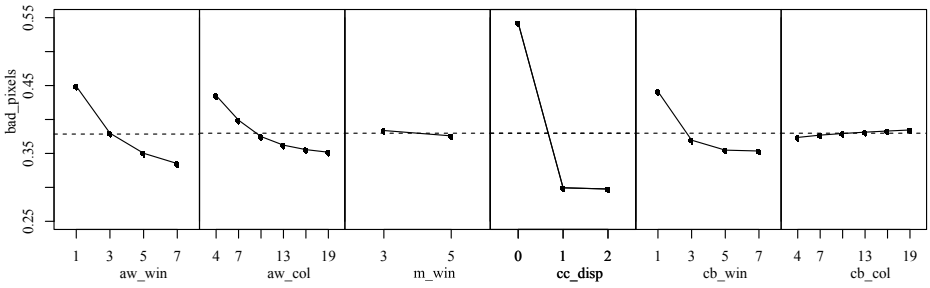


Fig. 3. Main Effects Plots of each factor level for all data sets. Steeper slopes relate to bigger influence on the variance of the *bad_pixels* output measurement.

The optimal settings in the original data set are presented in Table 3 along with the proposed combinations.

Run Type	**bad_pixels**	**aw_win**	**aw_col**	**m_win**	**cc_disp**	**cb_win**	**cb_col**
Low Settings	67.62%	1	4	3	0	1	4
High Settings	19.84%	7	19	5	2	7	19
Best Initial	16.78%	7	19	5	1	3	4
Exploratory analysis	14.48%	9	22	5	1	3	4
MVLR optimization	14.66%	11	22	5	3	3	18

Table 3. Model comparison. Average *bad_pixels* values over all data sets and their parameter settings. Low and high settings comprise the depth maps with all their parameter settings at each of their minimum and maximum tested values respectively. Best initial are the most accurate depth maps from the study data set. Exploratory analysis corresponds to the settings determined using the exploratory data analysis based on box plots and main effects plots. MVLR optimization is the extrapolation optimization of the classical data analysis based on multi-variate linear regression model, nested models, and ANOVA

5 Conclusions and Future Work

This work is presented as a structured methodology to measure the relative influence of the inputs of a depth map algorithm on the output variance and the identification of new settings to improve the results from 16.78% to 14.48% *bad_pixels*. The methodology is applicable on any group of depth map image sets

generated with an algorithm where the relative influence of the user-specified parameters merits to be assessed.

Using design of experiments reduces the number of depth maps needed to carry out the study when a large image database is not available. Further analysis on the input factors should be started with exploratory experimental fractional factorial designs comprising the full range on each factor, followed by a response surface experimental design and analysis. In selecting the factor levels, analyzing the influence of each filter independently would be an interesting criterion.

Acknowledgments. This work has been partially supported by the Spanish Administration Agency CDTI under project CENIT-VISION 2007-1007, the Colombian Administrative Department of Science, Technology, and Innovation; and the Colombian National Learning Service (COLCIENCIAS-SENA) grant No. 1216-479-22001.

References

1. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vision, 47(1-3):7–42 (2002)
2. Gong, M., Yang, R., Wang, L., Gong, M:. A performance study on different cost aggregation approaches used in real-time stereo matching. Int. J. Comput. Vision, 75:283–296 (2007)
3. Yoon, K., Kweon, I.: Adaptive support-weight approach for correspondence search. IEEE Trans. Pattern Anal. Mach. Intell., 28(4):650 (2006)
4. Congote, J., Barandiaran I., Barandiaran, J., Montserrat, T., Quelen, J., Ferrán, C., Mindan, P., Mur, O., Tarrés, F., Ruiz, O.: Real-time depth map generation architecture for 3d videoconferencing. 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2010, 1–4 (2010)
5. Gu, Z., Su, X., Liu, Y., Zhang, Q.: Local stereo matching with adaptive support-weight, rank transform and disparity calibration. Pattern Recogn. Lett., 29:1230–1235 (2008)
6. Hosni, A., Bleyer, M., Gelautz, M., Rhemann, C.: Local stereo matching using geodesic support weights. Proceedings of the 16th IEEE Int. Conf. on Image Processing (ICIP), 2093–2096 (2009)
7. Wang, L., Gong, M., Gong, M., Yang, R.: How far can we go with local optimization in real-time stereo matching. Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), 129–136 (2006)
8. Fua, P.: A parallel stereo algorithm that produces dense depth maps and preserves image features. Machine Vision and Applications, 6(1):35–49 (1993)
9. Weiss, B.: Fast median and bilateral filtering. ACM Trans. Graph., 25:519–526 (2006)
10. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. IEEE Conference on Computer Vision and Pattern Recognition, 1:195–202 (2003)
11. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. IEEE Conference on Computer Vision and Pattern Recognition, 0:1–8 (2007)
12. Hirschmuller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. IEEE Conference on Computer Vision and Pattern Recognition, 0:1–8 (2007)
13. Tombari, F., Mattoccia, S., Di Stefano, L., Addimanda, E.: Classification and evaluation of cost aggregation methods for stereo correspondence. IEEE Conference on Computer Vision and Pattern Recognition, 1–8 (2008)