STATISTICAL INPUT FACTOR EVALUATION IN AN ADAPTIVE WEIGHT DEPTH MAP ALGORITHM

STATISTICAL INPUT FACTOR EVALUATION 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 INPUT FACTOR EVALUATION 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

February 2011

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

EAFIT UNIVERSITY

COLLEGE OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

MEDELLIN

2011

Acknowledgments

This graduation project would not have been possible without the support and guidance of Dr. Prof. Eng. Oscar Ruiz. My gratitude goes to him, his knowledge and patience are key driving factors of the research conducted at the CAD CAM CAE Laboratory where the work summarized in this document was developed.

I also thank instrumental contributors to this work like advisor Dr. Prof. Eng. Diego Acosta who provided essential counsel on the statistical data analyses and tools, and advisor M. Sc. John Congote for sharing his own research results and giving constant 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.

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.

Table of contents

List of Figures

List of Tables

2 Statistical Input Factor Evaluation 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] *DPP 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 created a set of depth maps with multiple combinations of input factor levels. A methodology for the statistical evaluation of such data could support design decisions on further developments.

The content of this graduation project corresponds to the article of the same name by Diego Acosta, Alejandro Hoyos, John Congote, and Oscar Ruiz. 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
DPP 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

Abstract

The influence of window size, color factor, median window size, cross-check disparity delta, cross-bilateral window size, and cross-bilateral color factor on the quantitative accuracy evaluation of an adaptive weight depth map generation algorithm is evaluated to choose user-specified input factor values tuning the output of specific data sets. Depth maps computed with several factor settings are evaluated with the Middlebury Stereomatcher and analyzed using box plots, multi-variate regression, analyses of variance, and main effects plots. The contribution of each factor is assessed to aid in setting selected parameters, improving thereafter the results over each data set independently and over all at once to achieve a more robust algorithm.

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 with matching algorithms that generate a depth estimation at each image pixel to describe the relative distance of the surfaces on a scene.

Depth map algorithms work mostly with at least 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 by using procedures like those described in [2] to remove lens distortions and produce rectified images. This allows a direct comparison of pixel rows amongst images reducing the correspondence problem to a single dimension. The initial depth map can have some sparsely populated regions due to unmatched pixels or bad matches. Various filters are used to smooth such regions during post-processing. The algorithm and filters use several manually-adjusted parameters before generating the depth map of an image pair. A good setting for the user-specified factors' levels of a given depth map algorithm is heavily influenced by the specific data set evaluated [3]. Published articles usually report the values set for their specific case studies without explaining the procedure followed to fine tune such settings [4], [5], [6], and some explicitly state the empirical nature of the values [7]. The variation of the output as a function of several settings in the support window size and the color factor are briefly discussed while not taking into account the effect of simultaneously modifying all factors [3], [5], [8]. Multiple stereo methods are compared choosing values based on experiments, but only some of the algorithm components are changed without showing how to select and set those values [1]. The commonly used approaches in determining the values on depth map algorithm parameters show the following weaknesses: (i) undocumented procedures to aid in setting values on new implementations, (ii) lack of planning when testing for the best settings, and (iii) failure to consider the interactions of varying all the factors at the same time.

A procedure to fine-tune user-specified parameters of a depth map algorithm is presented. It is tested using a set of images from the adaptive weight technique results from the implementation in [4], the Middlebury Stereomatcher, and the R software environment for statistical computing. Multiple values are used on all parameters and evaluated to measure the contribution of each factor to the result variance. The quantitative accuracy evaluation allows the use of main effects plots and analyses of variance of multi-variate regression models, to aid in the selection of the best combination of values for each data set. The evaluation results achieved with the initial data set are then improved by setting new values on user-specified parameters and the algorithm is tuned to work well with new rectified image pairs.

This article is based on work done in [1] where the principles of the stereo correspondence techniques and the quantitative evaluator are discussed. Section 4 contains the general information on stereo correspondence, including adaptive weights, median smoothing, disparity cross check, and bilateral filtering. The experimental set-up is shown in Section 5. Section 6 displays the results.

4 Background

Disparity is commonly used to describe inverse depth in computer vision. It is also used to describe the perceived spatial shift of a feature observed from different but 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 generally use at least a pair of images taken using a known camera geometry producing 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 that features display a consistent appearance on slightly shifted camera viewpoints. The steps generally taken to compute the depth maps may include any sequence of: (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 optimizing a given criterion (i.e. minimize the matching cost). Adaptive weight is an implementation of a local depth map generation algorithm that matches costs over fixed moving windows.

4.1 Adaptive Weight Algorithm

Adaptive weight [5] is an implementation of a local algorithm where 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 its depth value, with higher values reflecting greater shifts and closer proximity to the camera.

The strength of grouping by color $(f_s(\Delta c_{pq}))$ for pixels p and q is defined by the Euclidean distance between colors (Δc_{pq}) by Equation 1. Similarly, grouping strength by distance $(f_p(\Delta g_{pq}))$

is defined by the Euclidean distance between pixel image coordinates (Δg_{pq}) by Equation 2.

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

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

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

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.

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

where $w\left(p, q\right) = f_s\left(\Delta c_{pq}\right) * f_p\left(\Delta g_{pq}\right)$, $\bar{p}_d$ 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 do most of their work on the matching cost computation and aggregation, estimating the final disparity of each pixel by selecting the minimum cost value with a winner take all optimization without any global reasoning after the dissimilarity computation.

4.2 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, areas with high variation in disparity, and occluded zones. The winner takes all approach only enforces uniqueness of matches for the reference image while points in the target image may get matched more than once, requiring checking the disparity estimates and filling any gaps with information from neighboring pixels.

Median filters are widely used in digital image processing to smooth signals at the expense of edge preservation, and can be applied to remove incorrect matches and holes by assigning neighboring disparities. 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 in 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 becomes another of the user specified parameters.

A definition of a valid disparity measure is proposed in [9] where the two images are both used as reference and target by cross-checking. 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.

The bilateral filter [10] 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.

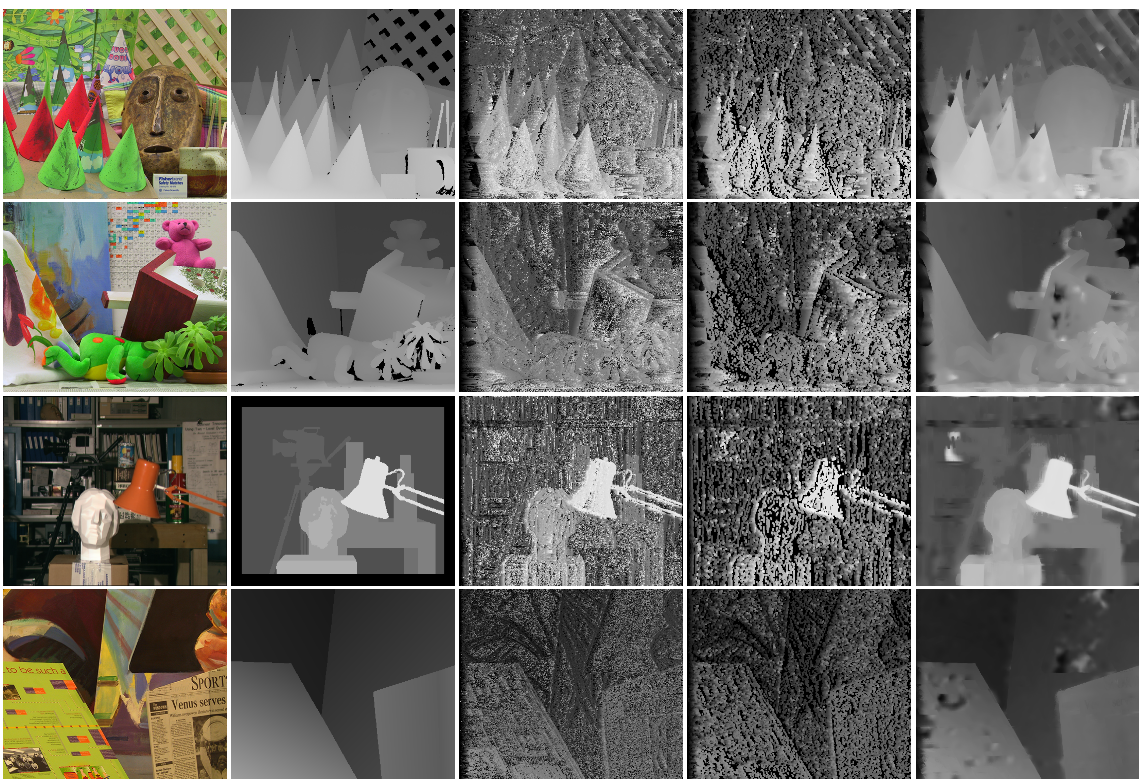
$$O_p = \frac{\sum_{q \in W} f_s\left(q-p\right) g_i\left(I_q - I_p\right) I_q}{\sum_{q \in W} f_s\left(q-p\right) g_i\left(I_q - I_p\right)} \tag{4}$$

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 Experimental

5.1 Data Sets

The depth maps are calculated with an implementation developed for real time videoconferencing [4] using well-known rectified image sets: Cones from [1], Teddy and Venus from [11], 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 [12] [13]. 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.



(a) (b) (c) (d) (e)

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.

A thorough understanding of the specific workings of the algorithm and its post-processing filters is needed to identify the input parameters that are subject to user modification when the performance is fine tuned. Each factor has a range of values that can be used to measure its influence. These values can be set at as many levels as desired as long as the difference is meaningful enough so that the change in input can be perceived later in the variation of the output. The input factors of the selected adaptive weight algorithm, its post processing filters, and the data sets are shown in Table 1. 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, facilitates the comparison of the contribution of each factor to the variation of the output.

Variable	**Name**	**Levels**	**Values**	**Coding**
Image Data Set		4	$[cones\ teddy\ tsukuba\ venus]$	
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 1: Data sets, algorithm, and filter user controllable factors with their evaluated value levels and codings.

Many recent stereo correspondence performance studies use the Middlebury Stereomatcher for their quantitative comparisons [3], [8], [14]. The evaluator code, sample scripts, and image data sets are available from the Middlebury stereo vision site [15], 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.

5.2 Evaluation

The evaluator takes as input the depth map in several formats, including portable network graphics (png), and a set of parameters related to the image sets. It outputs a text file with the root mean square errors (RMS) and proportion of bad pixels over typical problem regions as presented in Table 2. The quality of the computed correspondences is quantified with a performance evaluation based on 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. An algorithm may perform reasonably well if measured on bad pixels and yet have a large RMS error due to the poor matches in limited regions. RMS errors become relevant at very low bad pixel percentages [1]. The online Middlebury stereo evaluation table gives a visual indication of how well the methods perform using the average percentage of bad pixels (bad_pixels) over all pixels (bad_pixels_all), non-occluded pixels (bad_pixels_nonocc), and pixels 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 2: Result metrics computed by the Middlebury Stereomatcher evaluator.

5.3 Statistical Analysis

The algorithm, image, factor, and output data is analyzed with statistical tools using the R software environment for statistical computing and graphics [16]. The relations amongst inputs and amongst outputs are measured with correlation analysis, 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 models the output variable as a function of all the factors to find the equation coefficients, correlation of determination, etc.

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

where $\hat{y}$ is the predicted variable, x is the factor, and β is the coefficient.

The analysis of variance (ANOVA) of the linear model allows to ponder the relative influence of each factor on the response. These influences are also displayed with main effect plots. Residual analyses are checked to validate the assumptions of the model (i.e. constant error variance, mean of errors is zero) [17].

6 Results and Discussion

6.1 Correlation

The correlation tests on the input factors show they are independent and must be included in the evaluation. The Pearson correlation coefficient values for the outputs are shown in Table 3, where a strong correlation amongst bad_pixels and the other outputs is evident, allowing the selection of this output alone for the study.

6.2 Box Plots

The box plots presented in Figure 2 appear to show statistically significant lower bad pixels values from using filters instead of none, cross-check disparity delta values greater than 0, adaptive weight window sizes on the high end, and large adaptive weight color factor values. Median window size, bilateral windows size, and bilateral window color values do not show a significant statistical difference in their influence on the selected output at the studied levels.

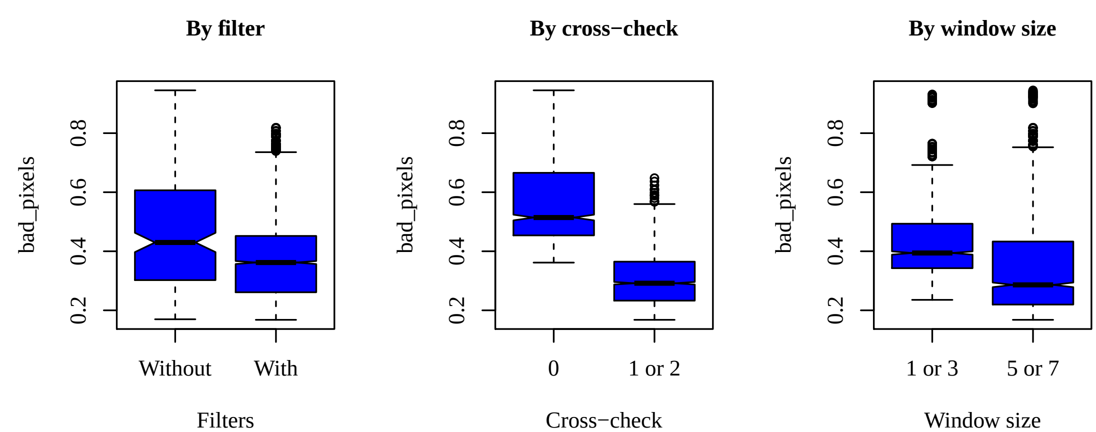


Figure 2: Box plots showing apparent differences on the bad_pixels output measure from grouped levels of algorithm and filter factors.

	bad_pixels	**rms_error_all**	**rms_error_nonocc**	**rms_error_occ**	**rms_error_textured**	**rms_error_textureless**	**rms_error_discont**	**bad_pixels_all**	**bad_pixels_nonocc**	**bad_pixels_occ**	**bad_pixels_textured**	**bad_pixels_textureless**	**bad_pixels_discont**
bad_pixels	1.00	***0.81***	***0.82***	0.59	***0.83***	***0.77***	***0.84***	***1.00***	***1.00***	***0.86***	***1.00***	***0.95***	***0.99***
rms_error_all	***0.81***	1.00	1.00	0.69	1.00	0.98	0.99	0.82	0.82	0.64	0.85	0.70	0.79
rms_error_nonocc	***0.82***	1.00	1.00	0.71	1.00	0.98	0.99	0.83	0.82	0.67	0.85	0.71	0.80
rms_error_occ	0.59	0.69	0.71	1.00	0.70	0.77	0.74	0.62	0.61	0.68	0.61	0.63	0.53
rms_error_textured	***0.83***	1.00	1.00	0.70	1.00	0.98	0.99	0.83	0.83	0.67	0.86	0.72	0.81
rms_error_textureless	***0.77***	0.98	0.98	0.77	0.98	1.00	0.98	0.78	0.78	0.64	0.80	0.68	0.73
rms_error_discont	***0.84***	0.99	0.99	0.74	0.99	0.98	1.00	0.85	0.84	0.67	0.87	0.73	0.82
bad_pixels_all	***1.00***	0.82	0.83	0.62	0.83	0.78	0.85	1.00	1.00	0.85	1.00	0.96	0.98
bad_pixels_nonocc	***1.00***	0.82	0.82	0.61	0.83	0.78	0.84	1.00	1.00	0.85	1.00	0.96	0.98
bad_pixels_occ	***0.86***	0.64	0.67	0.68	0.67	0.64	0.67	0.85	0.85	1.00	0.83	0.87	0.86
bad_pixels_textured	***1.00***	0.85	0.85	0.61	0.86	0.80	0.87	1.00	1.00	0.83	1.00	0.93	0.99
bad_pixels_textureless	***0.95***	0.70	0.71	0.63	0.72	0.68	0.73	0.96	0.96	0.87	0.93	1.00	0.93
bad_pixels_discont	***0.99***	0.79	0.80	0.53	0.81	0.73	0.82	0.98	0.98	0.86	0.99	0.93	1.00

Table 3: Pearson correlation coefficient for the evaluator outputs over all data sets.

6.3 Multi-Variate Regression

The formula, coefficients, goodness of fit, and analysis of variance for all data sets are presented in Table 4.

Call

lm(formula=bad_pixels∼ aw_win + aw_col + m_win + cc_disp + cb_win + cb_col)

Coefficients

	Estimate	Std. Error	t value	Pr(> $\|t\|$)	Signif.
(Intercept)	0.378524	0.001504	251.736	<2e-16	0
aw_win	-0.055691	0.002017	-27.606	<2e-16	0
aw_col	-0.040037	0.002201	-18.189	<2e-16	0
m_win	-0.003935	0.001504	-2.617	0.0089	0.001
cc_disp	-0.122413	0.001842	-66.471	<2e-16	0
cb_win	-0.041663	0.002017	-20.652	<2e-16	0
cb_col	0.005342	0.002201	2.427	0.0153	0.05

Residual standard error	0.0884	on 3449 degrees of freedom
Multiple R-squared	0.6331	
Adjusted R-squared	0.6324	
F-statistic	991.8	on 6 and 3449 degrees of freedom
p-value	<2.2e-16	

Analysis of Variance

	Df	Sum Sq	Mean Sq	F value	Pr(> F)	Signif.
aw_win	1	5.955	5.955	762.0978	<2.2e-16	0
aw_col	1	2.585	2.585	330.8483	<2.2e-16	0
m_win	1	0.054	0.054	6.8502	0.008902	0.001
cc_disp	1	34.525	34.525	4418.4094	<2.2e-16	0
cb_win	1	3.333	3.333	426.5237	<2.2e-16	0
cb_col	1	0.046	0.046	5.8899	0.015279	0.05
Residuals	3449	26.950	0.008			

Table 4: Multivariate Linear Model Regression Data

The model coded coefficients and goodness of fit measure for each data set are presented in Table 5 from which the analysis of variance is conducted to quantify the factors with the most influence as shown by the values in Table 6 and the slopes in the main effects plots of Figure 3.

The residuals of the models do not follow a normal distribution. Transforming the output variable, or removing large residuals does not improve the residual distribution as shown in Figure 4, and

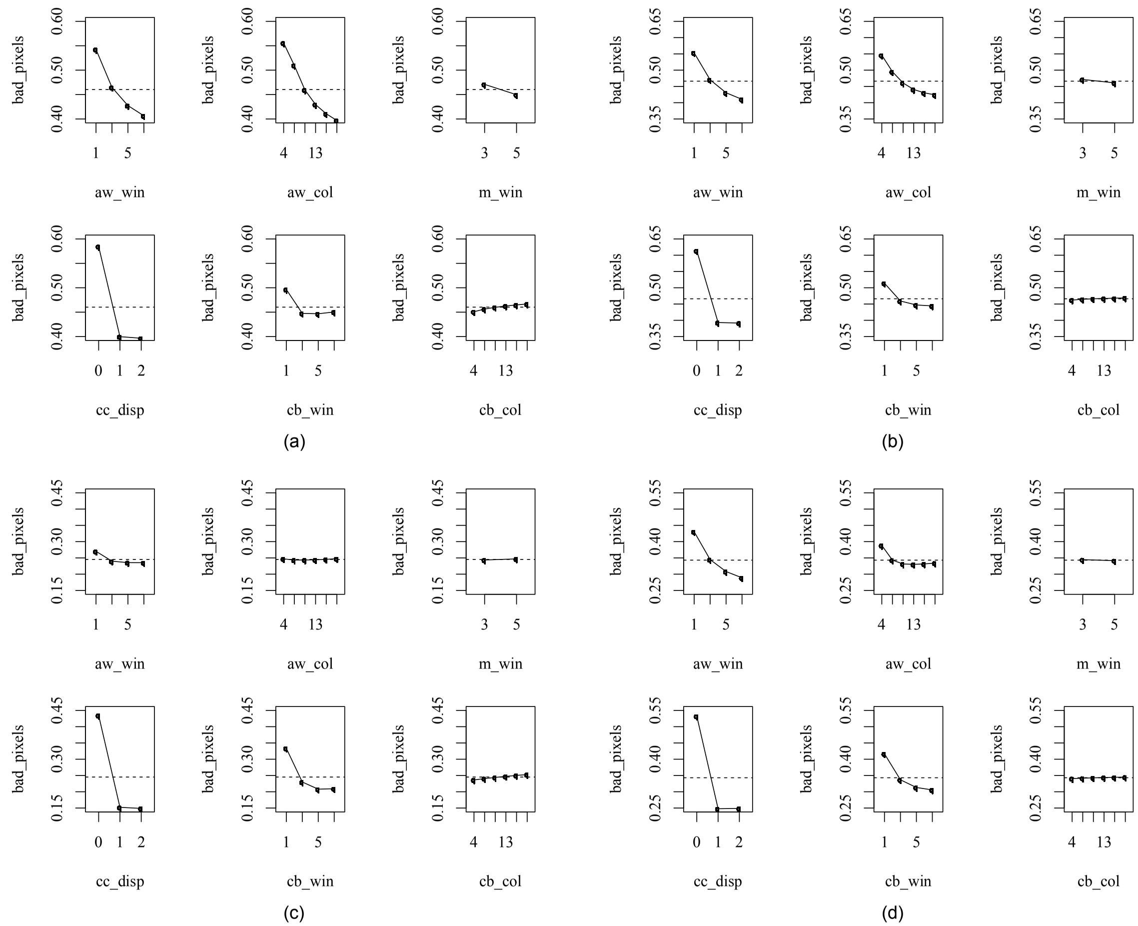


Figure 3: Main Effects Plots of each factor level per data set. (a) Cones, (b) Teddy, (c) Tsukuba, (d) Venus. Steeper slopes point to a bigger influence on the variation of the bad_pixels output measurement.

Data set	R^2	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_{12}	β_{56}
Cones	71.16%	+0.4602	−0.0667	−0.0800	−0.0106	−0.0939	−0.0208	+0.0077	−0.04221	+0.0050
Teddy	66.94%	+0.4660	−0.0704	−0.0589	−0.0052	−0.1111	−0.0332	+0.0031	−0.02233	+0.0015
Tsukuba	58.17%	+0.2451	−0.0164	+0.0007	+0.0016	−0.1429	−0.0591	+0.0082	+0.01186	+0.0071
Venus	63.19%	+0.3429	−0.0693	−0.0220	−0.0015	−0.1417	−0.0535	+0.0023	+0.00809	+0.0020
All	63.39%	+0.3785	−0.0557	−0.0400	−0.0039	−0.1224	−0.0417	+0.0053	−0.01114	+0.0039

Table 5: Linear model regression equations with goodness of fit measure and coded coefficients for each data set.

Data set	cc_disp	aw_win	aw_col	cb_win
Cones	34.35%	14.46%	17.47%	–
Teddy	41.25%	13.78%	8.10%	–
Tsukuba	50.24%	–	–	7.16%
Venus	47.35%	9.42%	–	5.62%
All	47.01%	8.11%	–	–

Table 6: Linear model ANOVA with the contribution to the sum of squared errors (SSE) of bad_pixels.

there are no apparent reasons to exclude any entries from the database for being outliers. The depth map images used in this study were not generated randomly, but their creation process is deterministic and should not show a different behavior if rerun randomly.

Some of the underlying assumptions of randomness, fixed distribution, fixed location, and fixed variation of the residuals are not satisfied after trying different approaches. Still, improved outputs could be found varying selected factors based on a faulty model.

The best combination of factor levels in the original data set are presented in Table 7 along with the proposed combinations. The output comparison is presented for each data set and for all of them at once to achieve an algorithm setting that performs reasonably well without being fine tuned to a specific image pair. The actual depth map outputs are shown in Figure 5.

The most noticeable influence on the output variable comes from having a cross-check filter to clear erroneous matches, accounting for nearly half the response variability in all the study data sets.

A cross-check filter with a disparity delta greater than 0 should be used first to improve the accuracy of an adaptive weight algorithm working with the same settings over multiple datasets, bigger window sizes can be applied next if the performance is not an issue, followed by increasing the color factor, and finally using bigger window sizes on the bilateral filter. Increasing the window

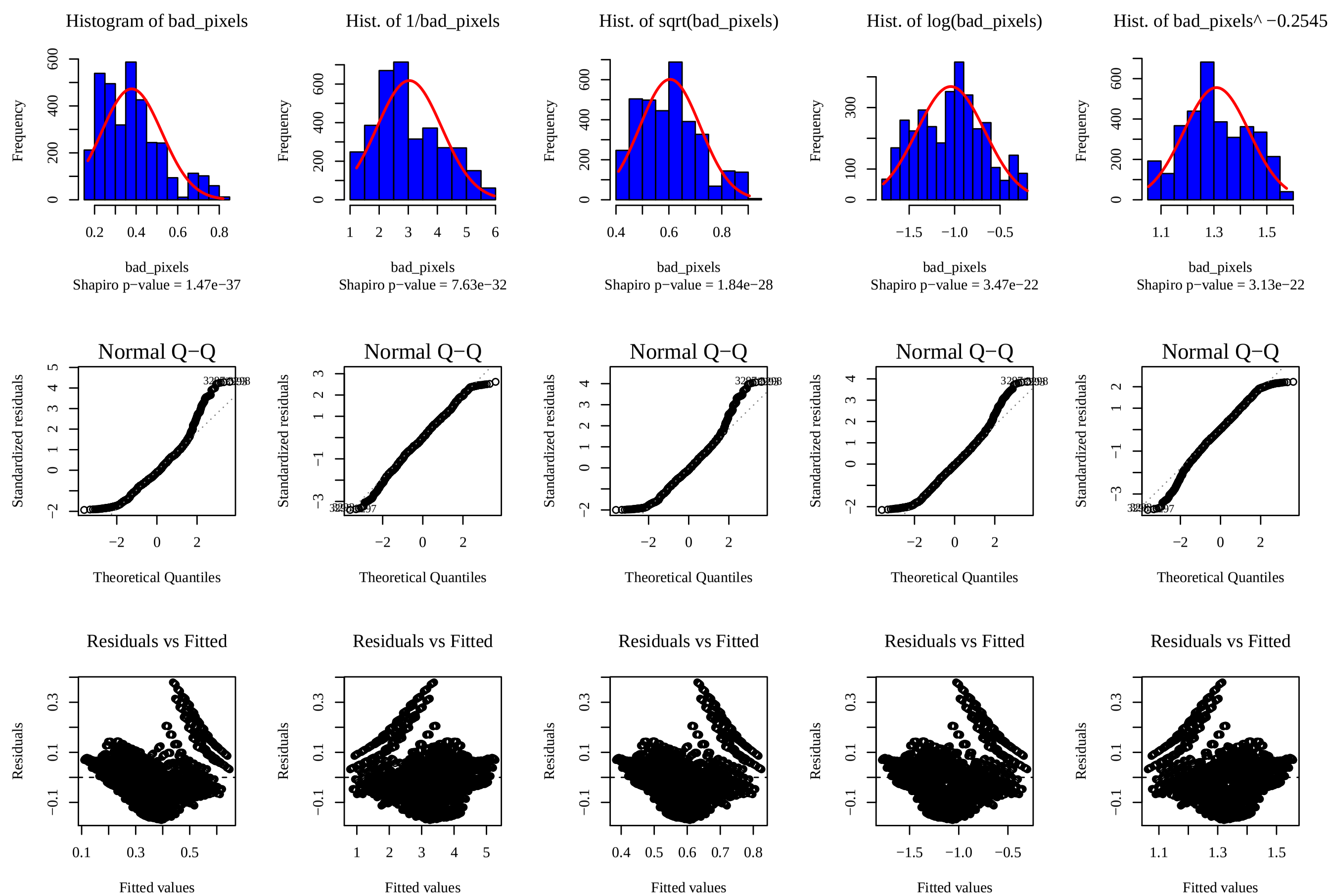


Figure 4: Output transformations. Top: Histogram, Middle: Quantile-Quantile Plot, Bottom: Residual versus Fits plots.

Data set	**Run Type**	bad_pixels	aw_win	aw_col	m_win	cc_disp	cb_win	cb_col
Cones	Best initial	20.76%	7	19	5	1	3	4
	Proposed	16.92%	9	22	5	1	3	4
Teddy	Best initial	22.81%	7	19	5	1	3	4
	Proposed	21.25%	9	22	5	1	5	4
Tsukuba	Best initial	9.68%	7	16	5	1	3	4
	Proposed	9.35%	9	22	5	1	5	4
Venus	Best initial	13.20%	7	16	5	1	7	4
	Proposed	10.37%	9	16	5	1	9	4
All	Best initial	16.78%	7	19	5	1	3	4
	Proposed	14.48%	9	22	5	1	3	4

Table 7: 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.

(a) (b) (c) (d)

Figure 5: Depth Map Comparison. Top: best initial, bottom: new settings. (a) Cones, (b) Teddy, (c) Tsukuba, and (d) Venus data set.

sizes on the main algorithm yield better results at the expense of longer running times, because 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 ratio between performance and accuracy.

7 Conclusions

This work used an existing database of non-randomly generated depth map images with multiple input factor values both in the adaptive weights algorithm and in the median, cross-check and bilateral post processing filters. The classical statistical approach using analyses of variance on multi-variate linear regression models alone is not suitable for this type of data analysis. Supporting the classical approach with box plots and main effects plots from the exploratory data analysis allows the identification of new values on selected factors to improve the desired results without considering the computational complexity.

8 Future Work

Future work related to the presented study could be directed at better understanding the good experimental results, as seen by the lower bad_pixels measurements on the new proposed runs, based on invalid statistical linear regression models.

Further analysis on the input factors should be started with exploratory experimental factorial designs with randomized runs and broader factor level ranges, and followed with a more focused experimental design and analysis.

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, and the CAD CAM CAE Laboratory at EAFIT University (COL).

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] 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.

[4] 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.

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

[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] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

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