Statistical tuning of Adaptive-Weight Depth Map Algorithm

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 systematic 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`. Our approach is systematic in the sense that the heuristics used for parameter tuning are supported by formal statistical methods. 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 surface-response designs 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 overall image processing 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 ([3]), 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

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.

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 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 shown 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.

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,

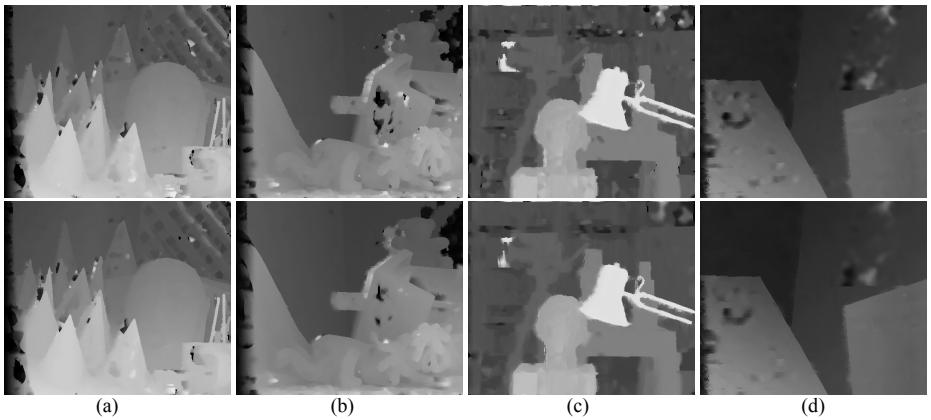


Fig. 1. Depth Map Comparison. Top: best initial, bottom: new settings. (a) Cones, (b) Teddy, (c) Tsukuba, and (d) Venus data set.

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

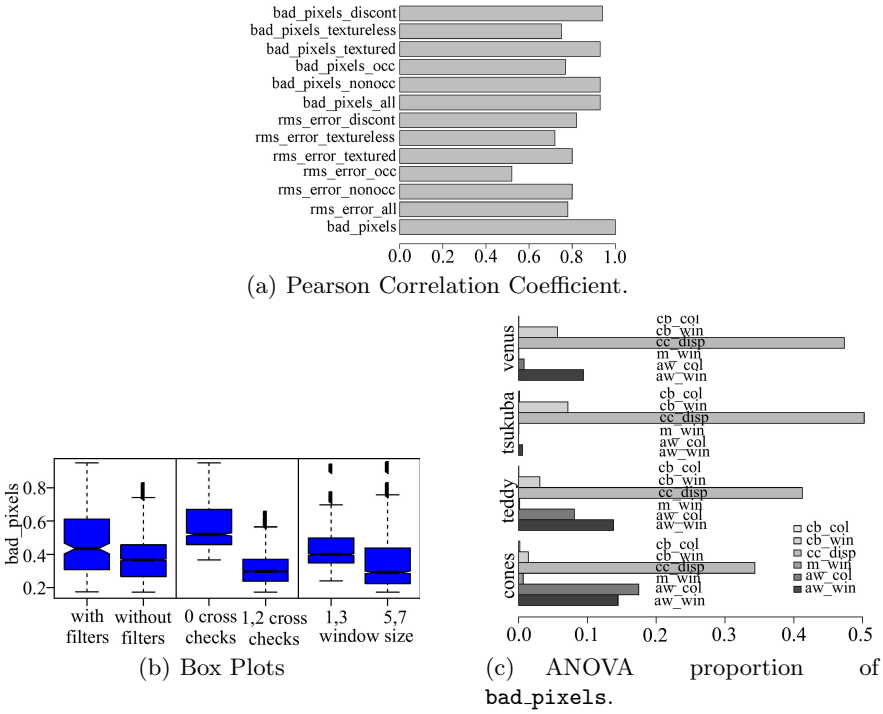
Variable selection. Pearson correlation of the factors show that they are independent and that each one must be included in the evaluation. On the other hand, a strong correlation amongst `bad_pixels` and the other outputs is detected and shown in Figure 2(a). This allows the selection of `bad_pixels` 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(b) 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%.

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

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.



(a) Pearson Correlation Coefficient.

(b) Box Plots

(c) ANOVA proportion of `bad_pixels`.

Fig. 2. (a) `bad_pixels` and other output correlation. (b) Box Plots of `bad_pixels`. (c) Contribution to the `bad_pixels` variance by parameter.

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 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. Window size is the next most influential factor, followed by color factor, and finally window size 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.

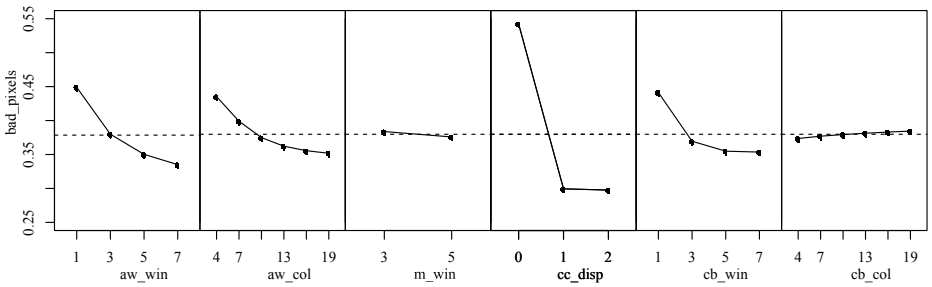


Fig. 3. Main Effects Plots of each factor level for all data sets. Steeper slopes relate to bigger influence on the variance of the `bad_pixels` output measurement.

The optimal settings in the original data set are presented in Table 3 along with the proposed combinations. **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 1.

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.

5 Conclusions and Future Work

This work presents a systematic 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)