Journal of Imaging Science and Technology® 55(5): 050201-1–050201-7, 2011.

Segment-Based Real Time Stereo Vision Matching Using Characteristic Vectors

Pablo Revuelta Sanz, Belén Ruiz Mezcua, and José M. Sánchez Pena
Carlos III University of Madrid, Spanish Center for Captioning and Audiodescription, Av. Peces Barba, 1. 28918 Leganés, Madrid, Spain
E-mail: prevuelt@ing.uc3m.es

Jean-Phillippe Thiran
Signal Processing Laboratory (LTS5), École Polytechnique Fédéral de Lausanne. (EPFL), Station 11, CH1015 Lausanne, Switzerland

Abstract. *Stereo vision is the normal method to obtain the depth information from images. The problems encountered when applying well established algorithms to real time applications are due to the high computational load required. In this article, the authors address this issue by performing a region-based analysis which considers each pixel only once. Additionally, matching is carried out over statistical descriptors of the image regions. In this article, the authors present a new algorithm that has been specifically designed to solve some commonly observed problems which arise from other well known techniques. This algorithm was designed using a previous algorithm implemented by the authors. The complete analysis has been carried out over gray scale images. The results obtained from both real and synthetic images are presented in terms of matching quality and time consumption and are compared with other published results. Finally, a discussion of additional features related to the matching process is provided.* © *2011 Society for Imaging Science and Technology.*
[DOI: xxxxxxxxxxxxxxx]

INTRODUCTION

Stereo vision is a process that provides three-dimensional perception by means of two different images of the same scene. This process is of great importance in computer vision, as it allows to obtain the distance from the cameras to any specific object of the scene. When different viewpoints from the same scene are compared, a further problem associated with the mutual identification of images arises. The solution to this problem is commonly referred to as matching. The matching process consists in identifying each physical point in different images.[1] The difference observed between images is referred to as disparity and allows depth information to be obtained from either a sequence of images or from a group of static images from different viewpoints.

In general terms, this approach solves the problem with local or global approximations. The first option takes into account only disparities within a finite window which presents similar intensities in both images.[2] Global algorithms make explicit smoothness assumptions converting the problem in an optimization one.[3] Among them, we can find some using edges, shapes and curves,[4–6] and points[7] or segment-based algorithms.[8] The calculations required for depth mapping of images have been studied in detail, and a complete review of algorithms performing this task by means of stereo vision can be found in Scharstein and Szeliski.[3]

The results presented in this article have been obtained using these processing techniques but especially taking into account scale and orientation information to accurately perform image matching.

Each of the three previously described approaches to the matching problem presents several computing problems. In the case of edges, curves and shapes, a differential operator must be used (typically Laplacian or Laplacian of Gaussian, as in Jia et al.[9]). This task requires a convolution of 3×3, 5×5 or even bigger windows; as a result, the computing load increases with the size of the operator (for separable implementations). This problem gets worse when using area-based matching algorithms; the computational load follows an exponential law. The use of a window to analyze and compare different regions seems to perform satisfactorily.[10] However, this technique requires many computational resources, as will be discussed later. Even most segment-based matching algorithms perform an $N \times N$ local windowing matching as a step of the final depth map computation.[3] It is important to notice that this step is not dimensional separable. Most of these algorithms, however, obtain very accurate results, with the counterpart of interpolating optimized planes to force solution of linear systems.[11]

In this article, we propose a novel matching algorithm based on characteristic vectors for grayscale images. The vectors are extracted with a region growing algorithm, taken from a previous work of the authors.[12] Likewise, a deeper study of the proposed algorithm has already been published by the authors as a technical report.[13]

This article follows the structure of the algorithm. Additionally, a section of results is presented with depth estimations of both real and synthetic standard images. An in-depth discussion and comparison with other algorithms is done in the light of the results and a final conclusion section closes this work.

Received Jan. 19, 2011; accepted for publication Jul. 17, 2011; published online Nov. 21, 2011
1062-3701/2011/55(5)/050201/7/$20.00.

PROBLEM STATEMENT

There are several geometrical and camera response assumptions that have been made to compare two images that are slightly different. These assumptions are described in detail by Pons and Keriven.[1] There are also various constraints that are generally satisfied by true matches thus simplifying the depth estimation algorithm, these are: similarity, smoothness, ordering, and uniqueness. All of these constraints are taken into account with the so called fronto-parallel and brightness constancy hypotheses.[1] Taking this into account, a region matching algorithm that reduces the number of operations needed for stereo matching is proposed, obtaining at the same time results that are relevant compared with those found in the bibliography.

The presented algorithm works as follows:

1. Image preprocessing. First of all, a Gaussian low pass filter is applied in order to reduce outlier pixels that are not representative. This task is crucial to perform the region growing algorithm that is described in the Section "Image preprocessing".
2. Characteristics extraction by region growing. Second, the region growing algorithm is applied and regions descriptors are obtained.
3. Vectors Matching. Once the vectors with the extracted descriptors are created, the matching process over the pair of vectors (one vector for each region and one array of vectors for each image) is implemented.
4. Depth Estimation. The depth estimation is computed from the horizontal distance of the centroid of every matched pair of region descriptors.

Image Preprocessing

The proposed algorithm has been designed to operate on grayscale images. Color images are first converted into 256 gray levels representing their brightness. Additionally, a smoothing filter is applied to reduce the influence of the noise on the processing. We use a 3×3 Gaussian filter.

The scope of the work presented in this part of the article is restricted to the fast segmentation of different regions and not coherent image segmentation (the fact that different segmented parts belong or do not belong to the same physical object is not of interest). Over-segmentation is then tolerated. An efficient method to set regions is done by truncating the image (and losing some information). In this implementation, this process is carried out by using the three most significant bits and, then, the grayscale is reduced to eight levels. The truncation process has been implemented on-the-fly inside the region growing and characteristics extraction section of the algorithm.

Characteristic Vectors for Region Labeling

Descriptors or characteristic vectors are extracted on-the-fly during region growing segmentation where each pixel is examined only once.[12] The most relevant characteristics obtained from each region are the area, gray value, centroid, length, width, boundary length, and orientation, being based on the most relevant visual cues used in human vision.[14]

Table I. Comparison tests and their corresponding thresholds.

Item	Characteristic	Comparison test	Acceptance thresholds
1	Centroid ordinates	Absolute difference	<(Image Height)/4
2	Centroid abscissa	Absolute difference and non-negative	[0, (Image Width)/4]
3	Value	Equality	1
4	Area	Relative difference	<55%
5	Length	Relative difference	<30%
6	Width	Relative difference	<30%
7	Angle	Weighed difference	<65
8	Vote of characteristics	Absolute addition	Total threshold

When the segmentation step is performed, we have a set of characteristic vectors describing each region of the image. In addition to these vectors, an image is required so that it maintains the reference between each pixel and the region identifier to which it belongs. This image is referred to as the "footprint image" and is composed of one byte per pixel, which represents the index of the characteristic region identifier in the vector. This limits, in this implementation, the number of segmented regions to 255 (the value "0" is reserved for unlabeled and occluded pixels). This kind of procedure is usually referred to as dense matching.[3]

Matching Based on Extracted Features

The matching process requires a specific characteristic that belongs to each of the different images and is obtained from different viewpoints. Using this novel algorithm, a chain of conditions has been proposed to verify the compliance between regions. With this structure, increased efficiency is achieved as every test is not always performed. The majority of the region characteristics are compared according to Eq. (1)

$$\mathrm{val} = \frac{\mathrm{abs}\left(\mathrm{Ch}^i_{\mathrm{left}} - \mathrm{Ch}^i_{\mathrm{right}}\right)}{\max\left(\mathrm{Ch}^i_{\mathrm{left}}, \mathrm{Ch}^i_{\mathrm{right}}\right)}, \tag{1}$$

where i represents the ith characteristic (Ch) of those presented in Table I of the left or right image.

For this case, the possible range of differences is [0, 1]. We refer to this particular operation as relative difference. Table I shows the order of conditions, the compared characteristic and the acceptance threshold.

The centroid coordinate matching in tests 1 and 2 is only searched in one-quarter of the image in each axis; it is assumed that all the potentially matched objects are located far enough from the cameras and in the same scan-line (1/4 image size vertical tolerance). Moreover, the difference of left and right centroid abscissas cannot be negative (which should represent objects far away from the infinite. This case is not taken into account since the specified geometrical assumptions are applied).

The preprocessed images, as said before, have been truncated, so pixel values must have the same values to be included in the same region, as in the third test.

The angle comparison (item number 7) needs deeper explanation. Because of the ambiguity of the angle measurement, when length and width are similar, a comparative function described in Eq. (2) has been implemented

$$\Delta\alpha = 100 \cdot \mathrm{abs}\left(\frac{2}{\pi} \cdot a\tan(\min(L_l - W_l, L_r - W_r)) \cdot \sin(\alpha_l - \alpha_r)\right). \tag{2}$$

In Eq. (2), L_x and W_x are the length and the width of the left and right image regions, respectively, and α_x, denoted by the subscript l and r are the angles of the left and right image region, respectively. By using this equation, the magnitude of the angle is high when there is a large difference between the length and width, and vice versa, because the angle measurement of a compact object (similar length and width) is highly noise sensitive.

Finally, if the result from all of the previous tests is positive, all of the differences obtained are added and compared with the sum of the applied thresholds. "Total_Threshold" is then computed in the first step by means of Eq. (3).

$$\text{Total\_Threshold} = \text{partial\_thresholds}. \tag{3}$$

After this operation (which is always positive in the first voting test), the result is stored and used as the new Total_Threshold value for further comparisons. By use of this procedure, if a further region is observed to fit more effectively into the reference region (i.e., the result from the addition is smaller), uniqueness of the matching function is enforced, and only this new region will be matched (and the previous region will be left unmatched).

This matching methodology has been implemented as a series of consecutive steps in a partial matching chain:

If some of the comparisons do not comply with the partial threshold, the inner loop is broken and reinitialized, saving computational load.

As stated in the introduction of this article, several geometrical assumptions have been considered, resulting in the following consequences:

- No geometrical correction is implemented. The two cameras are assumed to be parallel in orientation, and objects are far enough from the cameras. Then, only abscissa distortions are supposed to be perceived between both images.
- The depth map is approximated by parallel and non-overlapping planes.
- It has been assumed that every well-matched left centroid abscissa is higher than the right one (and they are equal when the region is located at infinity) and their difference lower than 25% of the image range. This means that the matching regions are assumed to be close to each other and, thus, located far enough from the cameras.
- Both images are taken from the same camera height, so the scan-line to find matches can be assumed to be horizontal where only a range of ±25% will be tolerated when searching for matches.
- No region with an area below 0.1% of the image size will be catalogued as a significant region and as a result will not be matched.

As the images projected on each of the camera planes are different, several of the areas in the scene might be projected on one of them, producing what is commonly referred to as the occlusion effect. These areas cannot be matched, as has been widely discussed in stereo vision literature.[15] It will be demonstrated that the method proposed in this article

```
for i=1…number_of_left_regions do
   total_Threshold = Σ default_partial_thresholds;          //partial thresholds from table II
   for j=1…number_of_right_regions do
     if 0 < (left_region[i].centroidX – right_region[j].centroidX) < width/4 then
       difY = abs(left_region[i].centroidY – right_region[j].centroidY); // difY is one of the partial_differences"
         if difY < height/4 then
           difArea = abs(left_region[i].area - right_region.area) ;   //difArea is one of the "partial_differences"
           if difArea < difAreaMax then                               // difAreaMax is one of the
                                                                      //"default_partial_differences"
               […]                                                    //This process is done for each characteristic
                 if Σ partial_differences < total_Threshold then
                   Match Found;
                   Total_Threshold = Σ partial_differences;
                   depth computation;
                 end if;
               […]
             end if;
           end if;
       end if;
     end for;
end for;
```

(a) (b)

Figure 1. (a) *Tsukuba* processed depth map. (b) True depth map.

leaves several regions where no matching occurs, and they will be indiscernible from occluded regions.

Regarding depth estimation, let *(x,y)* be a descriptor of the centroid of a region in the left image, and *(x',y')* the same descriptor of the right image. Taking into account, the geometrical assumptions detailed before, we can assume that

$$(x, y) = (x' + d \cdot x \pm \varepsilon_x, y' \pm \varepsilon_x), \tag{4}$$

where d is the distance (disparity) between the centroid abscissa, and ε the tolerance allowed in both directions.

Then, the main advantage of stereo vision is the correspondence between differences in the x axis, and the distance between the object and the cameras either once the cameras have been calibrated or the required assumptions have been made. The absolute distance of the centroid abscissas (in pixels) is measured for every matched pair of regions.

In this work, the left image is the reference and is used to compute the depth map. This is an arbitrary choice, without any loss in generality, but the Middlebury database forces this constraint to allow automatic error measurements.

RESULTS

The previously described algorithm has been implemented using the OpenCV library and tested over different standard stereo pairs of images. These tests have been implemented using a 1.6 GHz microprocessor. All the tested images encounter geometrical constrictions assumed by our algorithm, and were obtained from the Middlebury benchmark, with their truth depth map (taken from the Middlebury database[16]).

First of all, the proposed algorithm was run over the *Tsukuba* pair of images with a resolution of 288×384. In Figure 1, the left version image is the computed depth map and the right one the ground truth.

The original pair of images has been segmented into 102 and 97 regions. The errors in nonoccluded pixels, for a threshold of 2 (in absolute values), are 55.9%. The error for all pixels is 56.6% and the error in discontinuities is 66%. These results will be discussed in the Discussion section.

Regarding the computation time, the algorithm takes close to 24 ms for the segmentation process of each image. As shown in Revuelta Sanz et al.,[12] the segmentation time is quadratically related to the number of regions of the segmented images and directly related to the image size.

The characteristics vectors must be prepared and normalized, and several of the descriptors are computed after the image segmentation from the extracted data. The time taken in this task in the *Tsukuba* test is close to 90.5 μs for each one. This is not significant regarding the segmentation and characteristic extraction processes. Finally, the matching has been carried out over the computed vector and not over the original images. In this case, the proposed algorithm takes up to 700 μs to compare and match both vectors. We can see the total time consumed is around 50 ms (20 fps) for this stereo pair, which is within the real time constraint.

We have carried out other tests over highly textured images, using the *Teddy* and *Venus* grayscale images, of size 375×450. The corresponding results are presented in Figure 2.

For the *Teddy* pair of images, the error in nonoccluded pixels, all pixels and discontinuities are 79%, 80.7%, and 87%, respectively. For the *Venus* test, these errors are 73.9%, 74.2%, and 68.7%.

Computation times are 78.9 ms (12.7 fps) for the *Teddy* pair of images, and 76.6ms (13 fps) for the *Venus* test images.

DISCUSSION

The main contribution of the proposed algorithm is to solve the matching problem in stereo vision by comparing characteristics of regions instead of the regions themselves, reducing the computational cost, and paying the price of higher error rates. Since images are segmented into no more than 255 regions, the computational efficiency increases with the size of the image as opposed to both windowed-areas and visual cues methods.

The aim of our proposal is to retrieve depth map estimations with an important increase of the time performance, regarding other algorithms found in literature.

Truncation preprocessing has been presented apart from the algorithm, for the goal of better comprehension,

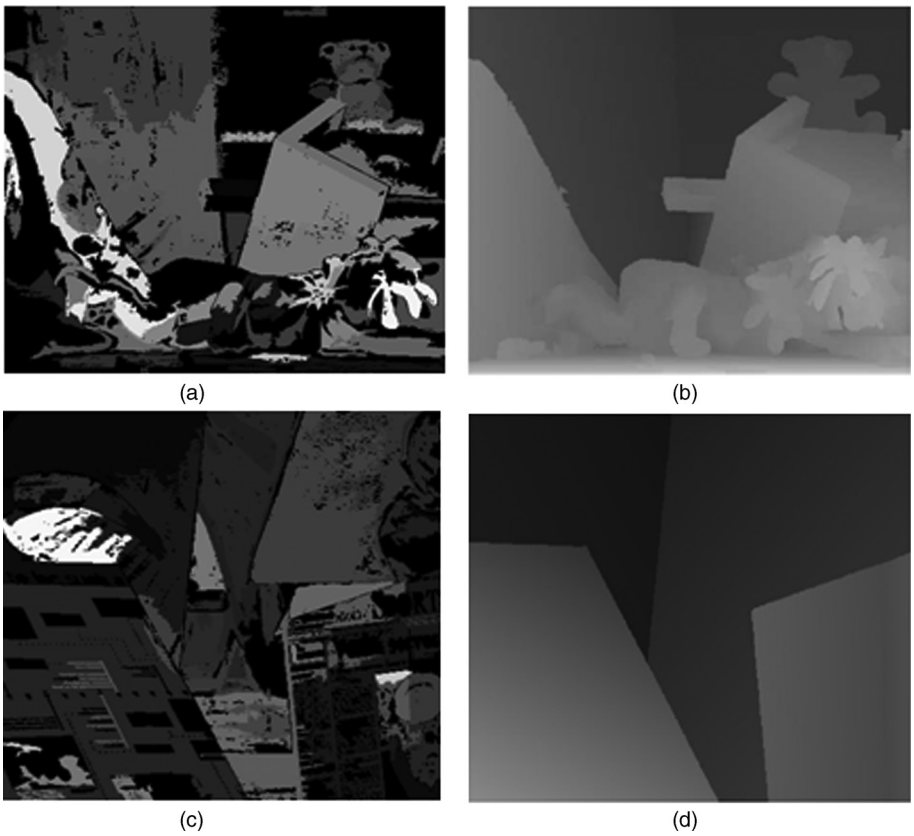

Figure 2. (a) *Teddy* computed depth map, (b) *Teddy* true depth map, (c) *Venus* computed depth map and (d) *Venus* true depth map.

but, in practice, it is implemented on-the-fly in the region growing algorithm. Thus, another advantage of the proposed algorithm is based on the fact that part of the preprocessing is carried out during the processing. No additional loops are then required in the program.

In contrast to other well-known methods which perform image matching,[10] we have replaced the task of comparing moving windows in both images by comparing two vectors that contain close to 100 terms in the *Tsukuba* images. The comparative process, due to its nested structure, allows time saving on many loops when certain tests have not been passed. However, we can obtain unexpected results when analyzing the higher time required to compute the matching of smaller images or lower number of segmented regions. The explanation of this possibility is based on the number of steps required to be carried out in the nested comparing structure. If the region descriptors are similar in values, the comparing structure needs to go through several levels, thus, generating an increased computational load for each vector.

The results obtained for the computed depth maps perceptually correspond to the truth depth map. However, errors are still evident: It can be seen that the gray scale segmentation based on brightness remains highly sensitive to noise. Other errors arise due to problems of matching small or undifferentiated figures such as the tins located behind the lamp in Fig. 1(a). When looking at quantitative results, we see high error rates. These errors are attributed to the following reason. The disparity is computed from the centroids differences. But such centroids are not always in the correct place, since some regions can "overflow" the physical region, including some extra pixels. This fact offsets the centroid abscissa and, hence, the final disparity is biased.

Figure 3 presents some results in (color based) depth map estimations from the same pair of images. These algorithms will be used for time performance comparisons.

The quality of these depth map estimations is higher than that proposed in this article, but regarding the time performance, we find some relevant data. Hirchsmüller et al. obtain their result with a 450MHz processor at 4.7 fps (note that the image size is 240×320 in this experiment).[17] Hong and Chen obtain the image in Fig. 3(b) with a 2.4GHz PC after 3 s (image size 288×384). In the case of Klaus et al., the computation time required is higher than 14 s on a 2.21GHz machine[11] (the image size is not specified in this work). We can easily see the improvement in terms of performance of our algorithm, since our results for the same image achieve a frame rate of 14 fps on a slower processor (except the case of Hirchsmüller et al.).

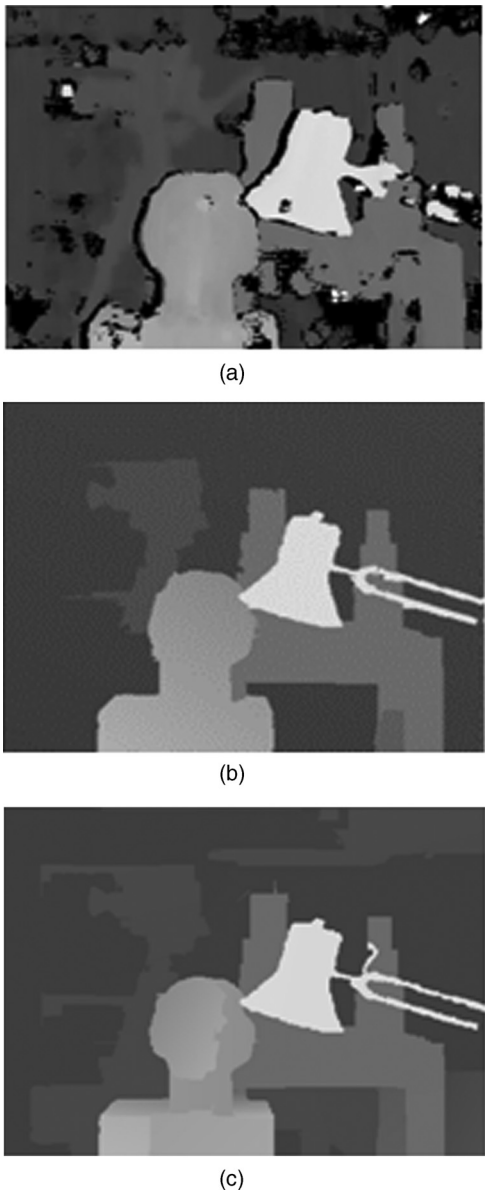

Figure 3. (a) Real time correlation based result taken from Hirchsmüller et al.,[17] (b) Segment-based result from Hong and Chen[8] and (c) another segment-based result from Klaus et al[11].

Another relevant point to be made is related to the unmatched and nonsegmented regions within the depth map (drawn in black). Most of the black regions are not segmented (i.e., when the area is smaller than the minimum allowed) and, hence, not matched. The errors then propagate from this discrimination to the final depth map. Other black regions belong to unmatched vectors. An example of this error is clearly presented in Fig. 2(c), where the left written panel has not been matched. These sets of images are rich in color and texture; their truncation into black and white produces a deficient segmentation and, thus, a deficient matching.

In the compared algorithms, the results obtained are very accurate, paying the price of high complexity and, hence, computational load. The scope of the research work presented in this article has enforced the implementation of a context-independent and faster algorithm resulting in a higher matching error rate. A balance must be reached between the segmentation, matching quality, and computational complexity. If a more accurate segmentation performance is required, color segmentation can be implemented; however, the main disadvantage of this process is that three times more calculations (one per channel) are required. In Bleyer and Gelautz,[10] using a 2GHz PC, researchers obtain for the *Tsukuba* (288×384) and the *Venus* (383×434) images, a computation time of 20 s. For the *Teddy* image (375×450), their algorithm takes around 100 s. Results are, again, much more accurate, but the computation time is between 400- and 1000-fold longer, depending on the image pair.

The main disadvantage of the present algorithm has been shown to be the dependence of the matching quality in terms of the region growing quality, as has been shown in Figs. 1 and 2, respectively, and their results.

CONCLUSIONS AND FUTURE WORK

A fast way of solving some problems of stereo vision has been proposed and tested in this work.

Our proposal can be summed up in the following aspects:

- Dense Matching: Theoretically, every pixel is linked to a region since the region growing algorithm processes all of them. In practical cases, some pixels remain unlabeled and, hence, unmatched.
- No interpolation: After the matching process no interpolation is done either over matched pixels (which would be nonsense) or over unlabeled pixels (to save computational load and tolerating unmatched regions).
- Real Time: All these aspects allow us to present a real time (around 14 fps) algorithm for large size images.
- Compromise accuracy-speed: The necessity has been shown for a compromise between accuracy (complexity) and the speed (computational load) of the algorithm, depending on the final application.

The algorithm presented in this work forms part of a first approach to the current problem of depth mapping and motivates further research to improve the segmentation process.

For future work, we would like to suggest the appropriateness of implementing motion estimation procedures in video sequences. Keeping track of unmapped pixels could drive higher frame rates and lower error levels. Moreover, some processing regarding color information should also be proposed, without trying to overload the global processing. However, we must take into account that every single step delays global performance.

ACKNOWLEDGMENTS

The authors would like to acknowledge the student grant offered by the Universidad Carlos III de Madrid and

Spanish Center for Captioning and Audiodescription (CESyA), which has allowed this research work to be performed.

REFERENCES

[1] J.-P. Pons and R. Keriven, "Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score", Int. J. Comput. Vis. **72**, 179 (2007).
[2] M. S. Islam and L. Kitchen, "Nonlinear similarity based image matching", Int. Fed. Inf. Process. **228**, 401 (2004).
[3] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", Int. J. Comput. Vis. **47**, 7 (2002).
[4] C. Schimd, A. Zisserman, and R. Mohr, "Integrating geometric and photometric information for image retrieval", Lect. Notes Comput. Sci. **1681**, 217 (1999).
[5] L. Szumilas, H. Wildenauer, and A. Hanbury, "Invariant shape matching for detection of semi-local image structures", Lect. Notes Comput. Sci. **5627**, 551 (2009).
[6] Y. Xia, A. Tung, and Y. W. Ji, "A novel wavelet stereo matching method to improve DEM accuracy generated from SPOT stereo image pairs", Int. Geosci. Remote Sens. Symp. **7**, 3277 (2001).
[7] J. Yu, L. Weng, Y. Tian, Y. Wang, and X. Tai, "A novel image matching method in camera-calibrated system", *2008 IEEE Conference on Cybernetics and Intelligent Systems* (IEEE, Piscataway, NJ, 2008), p. 48.
[8] L. Hong and G. Chen, "Segment-based Stereo matching using graph cuts", *Proc. 2004 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* (IEEE Computer Soc. Press, Los Alamitos, CA, 2004).
[9] Y. Jia, Y. Xu, W. Liu, C. Yang, Y. Zhu, X. Zhang, and L. An, "A miniature stereo vision machine for real time dense depth mapping", Lect. Notes Comput. Sci. **2626**, 268 (2003).
[10] M. Bleyer and M. Gelautz, "A layered stereo matching algorithm using image segmentation and global visibility constraints", ISPRS J. Photogramm. Remote Sens. **59**, 128 (2005).
[11] A. Klaus, M. Sormann, and K. Kraner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure", Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, Hong Kong. 15 (2006).
[12] P. Revuelta Sanz, B. Ruiz Mezcua, and J. M. Sánchez Pena, "Efficient characteristics vector extraction algorithm using auto-seeded region-Growing", *Proc. 9th IEEE/ACIS International Conference on Computer and Information Science* (IEEE, Piscataway, NJ, 2010) p. 215.
[13] P. Revuelta Sanz, B. Ruiz Mezcua, J. M. Sánchez Pena, and J.-Ph., Thiran, "Stereo vision matching using characteristics vectors EPFL-REPORT-150511", http://infoscience.epfl.ch/record/150511/files/TechReport_1.pdf accessed July 26, 2010.
[14] N. Ouerhani, A. Bur, and H. Hügli, "Linear vs. nonlinear feature combination for saliency computation: A comparison with human vision", Lect. Notes Comput. Sci. **4174**, 314 (2006).
[15] G. Egnal and R. P. Wildes, "Detecting binocular half-occlusions: Empirical comparisons of five approaches", IEEE Trans. Pattern Anal. Mach. Intell. 24 (8), 1127 (2002).
[16] See http://vision.middlebury.edu/stereo/.
[17] H. Hirchsmüller, P. R. Innocent, and J. Garibaldi, "Real time correlation-based stereo vision with reduced border errors", J. Comput. Vis. **47** (1/2/3), 229 (2002).