Chapter 2

Stereo Vision and Scene Segmentation

Carlo Dal Mutto, Fabio Dominio, Pietro Zanuttigh
and Stefano Mattoccia

Additional information is available at the end of the chapter

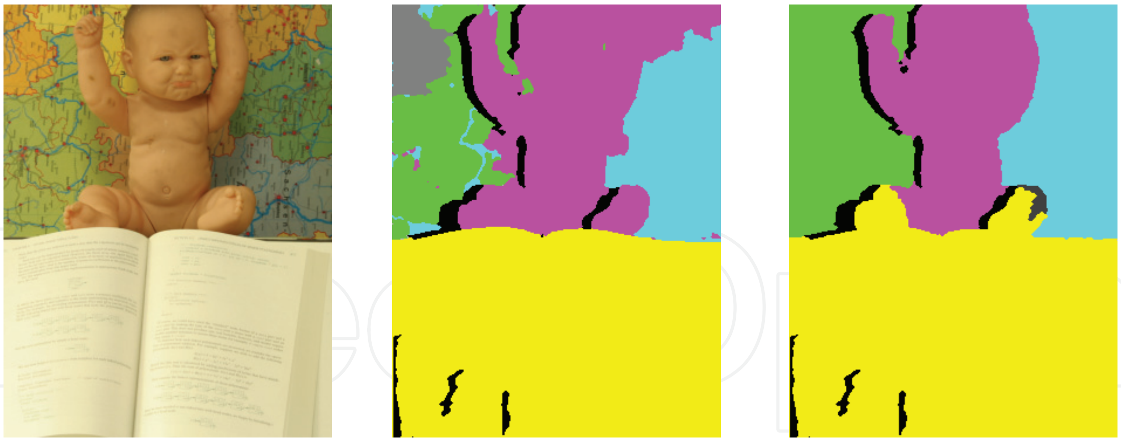
http://dx.doi.org/10.5772/45903

1. Introduction

Scene *segmentation* is the well-known task of identifying the image regions corresponding to the different scene elements or *segments* S_k, $k = 1, 2..., N$ belonging to a predefined set S partitioning the scene in N subsets, each one corresponding to a scene object or to a region of interest. Beside being an important problem by itself, segmentation is also employed as a preliminary step in many other computer vision tasks, e.g. object recognition or stereo vision. A closely related problem, very relevant for the television and movie industries, is *video-matting* which consists in separating the background from the foreground.

Classical segmentation techniques are based on different insights but all of them face the problem starting from the color information extracted from a single image of the framed scene. Despite the huge efforts put in research, scene segmentation from a single image is an ill-posed problem still lacking of robust solutions. The intrinsic limit of the classical approaches is that the color information contained in an image does not always suffice to completely understand the scene composition. An example of this limit is depicted in Figure 1(b). Note how the baby and part of the world map on background were associated to the same segment by a classical segmentation algorithm based on color only information, due to the very similar colors of the baby's skin and of some map regions.

Depth information can also be used for segmentation purposes. In this way some limits of color information based segmentation can be overcome but with side-effect of introducing other problems in regions that have similar depth but different colors. Figure 1(c) reports an example of this, as the book and the baby's feet were associated to the same segment due to their similar depth. The usage of geometry only allows good segmentation performance but is not always effective. For example it can not solve situations where there are objects of different colors placed close one to the other, such as two people wearing different clothes but very close each other, or slanted surfaces crossing multiple depths. At the same time color information only can not distinguish objects with similar colors regardless their relative distance.



(a) Color image

(b) Segmentation on the basis of color data

(c) Segmentation on the basis of geometry data

Figure 1. Results of a classical scene segmentation method applied to color or geometry information only. Occlusions or pixels discarded by segmentation algorithm are reported in black.

Clearly from Figures 1(b) and 1(c), segmentation based on either color or geometry information is likely to fail, since most of the framed scenes contain objects sharing similar colors or neighbors in the 3D space.

By considering both color and geometry cues in segmentation it is possible to avoid the ambiguities described above, as pixels with similar color but distant in 3D space or vice-versa are no longer likely to be mapped to neighboring feature vectors in feature space. It is also true that often the joint usage of color and geometry information as segmentation clues may be enough for this task. For example, if the algorithm of Figures 1(b) and 1(c) exploited both color and geometry, it would have "realized" that the baby's feet belong to the "baby segment" since they have the same baby's skin color though they share the same depth with the book, and that the map regions do not belong to the baby's segment as well though they share the same skin color, since they belong to the background and a have a different depth.

Many different approaches exist for the estimation of the 3D scene geometry. They can be roughly divided into active methods, that project some form of light over the scene, like laser scanners, structured light systems (including the recently released Microsoft's Kinect) or Time-Of-Flight cameras. Passive methods instead do not use any form of projection and usually rely only on a set of pictures framing the scene. In this class binocular stereo vision systems are the most common approach due to the simplicity of the setup and to the low cost. The choice of the best suitable system depends upon the trade-off among the system cost, speed and required accuracy.

Stereo vision systems provide estimates of the 3D geometry of a framed scene given two views of it, often referred as *left* and *right* view respectively. A considerable amount of research has been devoted for many years to scene geometry reconstruction by mean of stereo vision methods, now able to give dense and reliable depth information estimates.

Current literature [15] provides different algorithms to perform this task, each one with different trade-offs between reconstruction *accuracy* and *efficiency* (computational requirements). Simpler and faster stereo algorithms usually have poor accuracy, especially in presence of non-textured regions where the search for the correspondent points in the two images is likely to fail. More sophisticated algorithms (e.g. global algorithms), instead, allow in most cases better accuracy at the expense of higher computational costs.

This chapter focuses on how segmentation robustness can be improved by 3D scene geometry provided by stereo vision systems, as they are simpler and relatively cheaper than most of current range cameras. In fact, two inexpensive cameras arranged in a rig are often enough to obtain good results. Another noteworthy characteristic motivating the choice of stereo systems is that they both provide 3D geometry and color information of the framed scene without requiring further hardware. Indeed, as it will be seen in following sections, 3D geometry extraction from a framed scene by a stereo system, also known as *stereo reconstruction*, may be eased and improved by scene segmentation since the correspondence research can be restricted within the same segment in the left and right images.

The chapter is organized as follows: Section 2 presents an overall synthesis of the stereo vision and clustering methods considered for the proposed scene segmentation framework. The framework is instead illustrated in Section 3. Section 4 provides a comprehensive set of experimental results for scene segmentation of different datasets exploiting the different combinations of stereo reconstruction and segmentation algorithms. Section 5 finally draws the conclusions.

2. Related works

Current section shortly resumes the state-of-the-art stereo vision and segmentation methods considered in next sections, highlighting their qualities and flaws. Exhaustive surveys of stereo vision algorithms can be found in [2], [13] and [15]. Recent segmentation techniques are based on graph theory (e.g. [5]), clustering techniques (e.g. [3, 14]) and many other techniques (e.g. region merging, level sets, watershed transforms and many others). A complete review can be found in [15].

2.1. Stereo vision algorithms

The stereo vision algorithms that have been used inside the proposed framework are here briefly described.

Fixed Window

The Fixed Window (FW) algorithm is the basic local approach. It aggregates matching costs over a fixed square window and uses, as most local algorithms, a simple winner-takes-all (WTA) strategy for disparity optimization. Similarly to most local approaches, the aggregation of costs within a frontal-parallel support window implicitly assumes that all the points within the support have the same disparity. Therefore, FW does not perform well across depth discontinuities. Moreover, as most local algorithms, FW performs poorly in textureless regions. Nevertheless, thanks to incremental calculation schemes [4, 10], FW is very fast. For this reason, despite its notable limitations, this algorithm is widely used in practical applications. In the implementation proposed in this chapter, the cost function is the Sum of Absolute Differences (SAD).

Adaptive Weights

The AdaptiveWeights (AW) algorithm [17] is a very accurate local algorithm that uses a fixed squared window but weights each cost within the support window according to the image content. The weights are first computed, on the left and right image, similarly to a bilateral filter (i.e. deploying a spatial and a color constraint) and then multiplied to obtain a symmetric weight assigned to each cost within the support window. This method uses the WTA strategy for disparity optimization and the sum of Truncated Absolute Differences metric for the costs. AW provides very accurate disparity maps and preserves depth discontinuities. However, as for other local approaches, this method performs poorly in textureless regions. Moreover, the support windows shrinks to a few points (or equivalently, AW sets very small weights for several points) in presence of highly textured regions making this method error prone. The AW algorithm is computationally expensive: it requires minutes to process a typical stereo pair (the authors report 1 minute for small size images).

Segment Support

The Segment Support (SS) algorithm [16] is a local algorithm that aims at improving the AW approach by explicitly deploying segmentation. Similarly to AW, it aggregates weighted costs within a square support window of fixed size. Starting from the stereo pairs and the corresponding segmented stereo pairs, SS computes the weights on each image according to the following strategy: the weights of the points belonging to the same segment in which the central points lies is set to 1. The weight of the points outside such a segment are set according to color proximity constraint only and discarding the spatial proximity constraint. The overall weight assigned to each point is computed similarly to AW. In [16] it was shown that this strategy allows SS to improve the effectiveness of AW near depth discontinuities and in presence of repetitive patterns and highly textured regions. However, similarly to other local approaches, this method performs poorly in textureless regions. Although the segmentation of the stereo pairs can be quickly performed, SS has an execution time higher than AW. However, in [8] was proposed a block-based strategy referred to as FSD (Fast Segmentation-driven), inspired by [9], that enables to obtain equivalent results in a fraction of the time required by SS. It is very interesting to apply SS or FSD in the segmentation framework of Figure 2, because there is a segmentation step both before computing disparity and after the stereo matching calculation. In this work experimental results are reported only with SS.

Fast Bilateral

The Fast Bilateral Stereo (FBS) approach [9] combines the effectiveness of the AW approach with the efficiency of the traditional FW approach enabling results comparable to AW much more quickly. In this algorithm the weights are computed on each image and on a block basis with respect to the central point according to a strategy similar to AW. The weight assigned to each block is related to the difference between the color intensity of the central point and the average color intensity of the block. The costs within each block are computed, very efficiently, on a point basis by means of incremental calculation schemes. Therefore, at each point within a block, this method assigns the same weight and its point-wise matching cost. Disparity optimization is based on the WTA strategy. With block of size 3×3, FBS obtain results comparable to AW, well preserving depth discontinuities, in a fraction of the time required by AW. Increasing the block size decreases the accuracy of the disparity maps but reduces the execution time further. Moreover, in [9] it was shown that computing weights on block basis makes this method more robust to noise compared to AW. Similarly to other local algorithms described so far, FBS performs poorly in textureless regions.

Semi-Global Matching

The Semi Global Matching (SGM) algorithm [7] explicitly models the 3D structure of the scene by means of a point-wise matching cost and a smoothness term. However, this method is not a traditional global approach since the minimization of the energy function is computed, similarly to Dynamic Programming or Scanline Optimization approaches, in a 1D domain [13]. That is, several 1D energy functions computed along different paths are independently and efficiently minimized and their costs summed up. For each point, the disparity corresponding to the minimum aggregated cost is selected. In [7] the author proposes to use 8 or 16 different independent paths. The SGM approach works well near depth discontinuities, however, due to its (multiple) 1D disparity optimization strategy, produces less accurate results than more complex 2D disparity optimization approaches. Despite its memory footprint, this method is very fast (it is the fastest among the considered algorithms) and potentially capable to deal with poorly textured regions.

Stereo Graph Cut

The Graph Cut stereo vision algorithm (GC) introduced in [1] is a global stereo vision method. It explicitly accounts for depth discontinuities by minimizing an energy function that combines a point-wise matching cost and a smoothness term. The GC algorithm models the 3D scene geometry with a Markov random field in a Bayesian framework and determines the stereo correspondence solving a labeling problem. The energy function is represented as a graph and its minimization is done by means of Graph Cut, an efficient algorithm that relies on the Min-Cut/Max-Flow theorem. As most global methods, GC is computationally expensive and has a large memory footprint. However, as most global algorithms, it can deal with depth discontinuities and textureless regions.

2.2. Segmentation methods

Three different clustering schemes have been considered:

Segmentation by k-means clustering

K-means is a classical central grouping clustering algorithm. It is very simple to implement and it is pretty fast. It is not very precise when applied to scene segmentation, because it assumes that the distribution of the considered feature vectors $\mathbf{p}_i$ representing the points $p_i, i = 1, ..., N$ is a mixture of Gaussians. This assumption is not generally verified in the scene segmentation context and for this reason this clustering method applied to the set $\mathcal{V}$ may give poor results.

Segmentation by mean-shift

The mean-shift algorithm [3] is a standard non-parametric feature-space analysis technique exploitable as a clustering algorithm. It aims at locating the maxima of a density function, given some samples drawn from the density function itself. It is useful for detecting the modes of a density, and therefore for clustering the feature vectors in a very efficient way. Mean-shift clustering is very fast, but prone to return outliers. However it is worth considering this clustering technique since it is very fast, quite reliable and widely used in computer vision and image analysis.

Segmentation by spectral clustering with Nyström method

This method, proposed in [14], is a state-of-the-art clustering algorithm based upon pairwise affinity measures computed between all possible couples of points in $\mathcal{S}$. It does not impose any model or distribution on the points $p_i, i = 1, ..., N$, and therefore its results in practical situations are more accurate and robust than those of k-means and mean-shift. Spectral clustering alone is very expensive for both CPU and memory resources. This characteristic is intrinsic to the nature of the algorithm, because the computation of a pairwise affinity measure between all the points $p_i \in \mathcal{S}$ requires to build a graph that has a node for each point and an edge between each couple of points. Such graph is usually very large. However, one may obtain an approximated version of such a graph by imposing that not all the points are connected. The Nyström method, proposed in [6], is a way to approximate the graph, based on the integral eigenvalue problem. Spectral clustering with Nyström method provides a nice framework to incorporate the fact that $\mathcal{S}$ has to be partitioned into subsets where color and 3D geometry are homogeneous. The resulting speed of spectral clustering with Nyström method is comparable with the ones of k-means and mean-shift.

Table 1 roughly represents the differences in *accuracy* (final segments) and *efficiency* (execution time) among the considered segmentation methods.

Method	Accuracy	Execution time
K-means	Low	Fast
Mean-Shift	Good	Fast
Spectral Clustering	High	Slow
Spectral clust. with Nyström method	High	Fast

Table 1. Accuracy vs. efficiency of the considered segmentation methods.

3. Proposed framework

The goal of the proposed scene segmentation framework, as already stated in the introduction, is to perform scene segmentation by exploiting both 3D geometry and color information acquired by a stereo vision system. The proposed segmentation scheme encompasses three main steps:

a) a stereo vision reconstruction algorithm in order to compute the 3D geometry of the framed scene;

b) a way of jointly representing 3D geometry and color information;

c) a suitable clustering technique.

The segmentation pipeline may be subdivided into four main steps, listed below, and starts with acquisition of two views of the same scene acquired by a standard stereo setup. A more detailed description of each step is reported in the following paragraphs.

1) Estimation of the 3D scene geometry by a stereo vision algorithm;
2) construction of a new scene representation that jointly considers both geometry and color information;
3) application of a clustering algorithm on the combined color and geometry data;
4) final refinement stage in order to remove artefacts due to noise or errors in the geometry extraction.

The scheme in Figure 2 shows a detailed overview of the architecture of the proposed scene segmentation framework. Note how the scheme is a general framework inside which different stereo vision and segmentation algorithms can be fitted. The proposed scheme refers to the segmentation of the left image since the left camera is usually chosen as reference for the stereo system; the segmentation of the right image can be computed by just swapping the role of the two images in the proposed scheme.

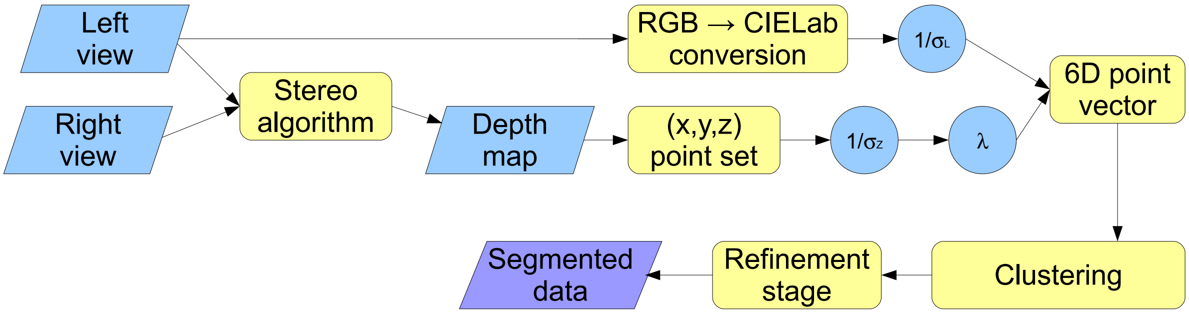


Figure 2. Architecture of the proposed scene segmentation method.

3.1. Estimation of the 3D geometry

In this first step the couple of images acquired by the calibrated and rectified stereo vision setup is given as input to a stereo vision algorithm in order to obtain the depth information associated to the framed scene points. It is possible to use any of the algorithms of Section 2 or any other available stereo vision algorithm.

Different stereo vision algorithms produce different depth maps (in terms of estimation accuracy) of the scene for the same input, and such differences may have a strong impact on the segmentation. Some examples of generated depth maps from different scenes (datasets) by the selected stereo vision algorithms are comparable in Figures 3, 5 and 7.

With the exception of GC, in the proposed implementations of all the considered algorithms there is a standard sub-pixel refinement step based on the fitting of a parabola in proximity of the best disparity.

3D geometry reconstruction, namely the computation of the coordinates (x, y, z) of the framed scene points, is performed by back-projecting the 2D undistorted coordinates $p_{2D,i} = (\tilde{u}_i, \tilde{v}_i, 1)$ of image lattice on the 3D space XYZ through Equation (1). Note how this process exploits the depth map produced by the chosen stereo algorithm and the stereo vision system parameters (intrinsic and extrinsic parameters of the two cameras forming the stereo pair).

$$\begin{bmatrix} x(p_i) \\ y(p_i) \\ z(p_i) \end{bmatrix} = z(p_i) K_S^{-1} p_{2D,i} \tag{1}$$

where K_S contains the intrinsic parameters matrix of the rectified stereo vision system (usually the intrinsic parameters of left camera).

Note how the occlusions are explicitly computed by cross-checking the disparity maps computed according to the reference and target images and the occluded points are discarded in the segmentation step.

3.2. Construction of the feature vectors

The geometrical description of the scene obtained in the previous step is now combined with color information in order to obtain better results than using geometry or color information only for the further segmentation, as stated before.

In order to exploit both types of information at the same time it is first of all necessary to build a unified representation that includes both color and 3D geometry data. Given a scene $\mathcal{S}$, after applying one of the stereo algorithms both 3D geometry and color information are available for all the scene points $p_i \in \mathcal{S}, i = 1, ..., n$ visible in both images (non occluded points in the stereo vision system field of view).

All such points can be represented by 6-dimensional vectors

$$\mathbf{V}_i = [L(p_i), a(p_i), b(p_i), x(p_i), y(p_i), z(p_i)]^T$$

where the first three components of $\mathbf{V}_i$ represent color information and the other three components represent geometry. The color information vector is built as follows: first of all the available color values are converted from the RGB to the CIELab *uniform* color space. A uniform color space, in fact, ensures that the Euclidean distance between points is close to the perceptual difference between the various colors and allows to compare the distances in the three color channels.

The 3D geometry information of each scene point p_i is represented, instead, by the 3D vector

$$[x(p_i), y(p_i), z(p_i)]^T$$

containing the point position in the three dimensional space.

Note how feature vectors V_i are not "clusterable" yet, since they are made by data of different nature (color and geometry) and magnitude, and segmentation methods require homogeneous feature vectors, that is vector components do have to belong to the same domain. Moreover, most of the mentioned methods require feature values to belong to $[0,1]$ range for a better operation.

For these reasons, after representing each point p_i by its 3D coordinates $x(p_i)$, $y(p_i)$ and $z(p_i)$ and color values $L(p_i)$, $a(p_i)$, $b(p_i)$, the proposed method applies a normalization to the resulting feature vectors. More precisely, Euclidean coordinates are normalized by the standard deviation σ_z of the z coordinate[1] and color information is normalized by the standard deviation σ_L of the L component.

Finally, the trade-off between the relevance of color and depth information is controlled by a factor λ. The final representation of each non-occluded point $p_i, i = 1, ..., N$ is then the 6-dimensional vector $\mathbf{p}_i, i = 1, ..., N$, defined as in Equation 2.

$$\mathbf{V}_i \triangleq \begin{bmatrix} \bar{L}(p_i) \\ \bar{a}(p_i) \\ \bar{b}(p_i) \\ \lambda\bar{x}(p_i) \\ \lambda\bar{y}(p_i) \\ \lambda\bar{z}(p_i) \end{bmatrix} = \begin{bmatrix} L(p_i)/\sigma_L \\ a(p_i)/\sigma_L \\ b(p_i)/\sigma_L \\ \lambda x(p_i)/\sigma_z \\ \lambda y(p_i)/\sigma_z \\ \lambda z(p_i)/\sigma_z \end{bmatrix}, i = 1, ..., N \tag{2}$$

It is evident from (2) that high values of λ raise geometry importance, while low values favor color information.

3.3. Segmentation

The result of the previous step is the set $\mathcal{V}$ of the 6D normalized vectors $\mathbf{p}_i, i = 1, ..., N$ describing the framed scene $\mathcal{S}$ taking into accounts for both geometry and color information. Assuming the scene $\mathcal{S}$ made by different meaningful parts $\mathcal{S}_k$, $k = 1, ..., K$, such as different objects or regions of interest, and recalling that segmentation is the task of finding the different groups of points representing the different objects, in the proposed framework segmentation can be formulated as the problem of clustering the vectors $\mathbf{p}_i, i = 1, ..., N \in \mathcal{V}$ into the clusters $\mathcal{V}_i, i = 1, ..., K$ representing the various objects. Each segment is so associated to a single cluster by using any of the clustering techniques described in Section 2.1. Note how the estimated depth maps can contain artifacts due to the limitations of the employed stereo vision algorithms and different combinations of stereo vision algorithms and clustering techniques can lead to different results.

Finally a refinement stage can be introduced in order to reduce the artifacts in the computed segmentations. A common post-processing step consist in looking for connected components and remove the ones with a size below a pre-defined thresholds. This allows to remove small artifacts typically due to noise in the images or samples with a wrongly estimated depth value. The samples in the removed regions are usually associated to the closest segment.

[1] The z axis is assumed to be parallel to the optical axis in order to make $z(p_i)$ correspond to the depth of the point p_i

4. Experimental results

In order to assess the feasibility of the approach some sample scenes have been segmented with different combinations of stereo vision and clustering techniques. This section shows the results of the performed tests.

In particular all the combinations of stereo vision and clustering algorithms of Section 3 have been tested on various scenes from the standard Middlebury dataset [11] and on scenes acquired in the *Multimedia Technology and Telecommunications Laboratory* (LTTM) of the University of Padova. Experimental results reported below are referred to the execution of Matlab implementations of clustering techniques. For what concerns the stereo vision algorithms we used our own C implementations except for Graph Cut and Semi-global Matching for which the implementations provided by the OpenCV library [12] have been used. The camera parameters of the Middlebury dataset are not available, hence they have been estimated in order to obtain a realistic 3D reconstruction from the ground truth. All the stereo vision system parameters for the LTTM dataset are instead known.

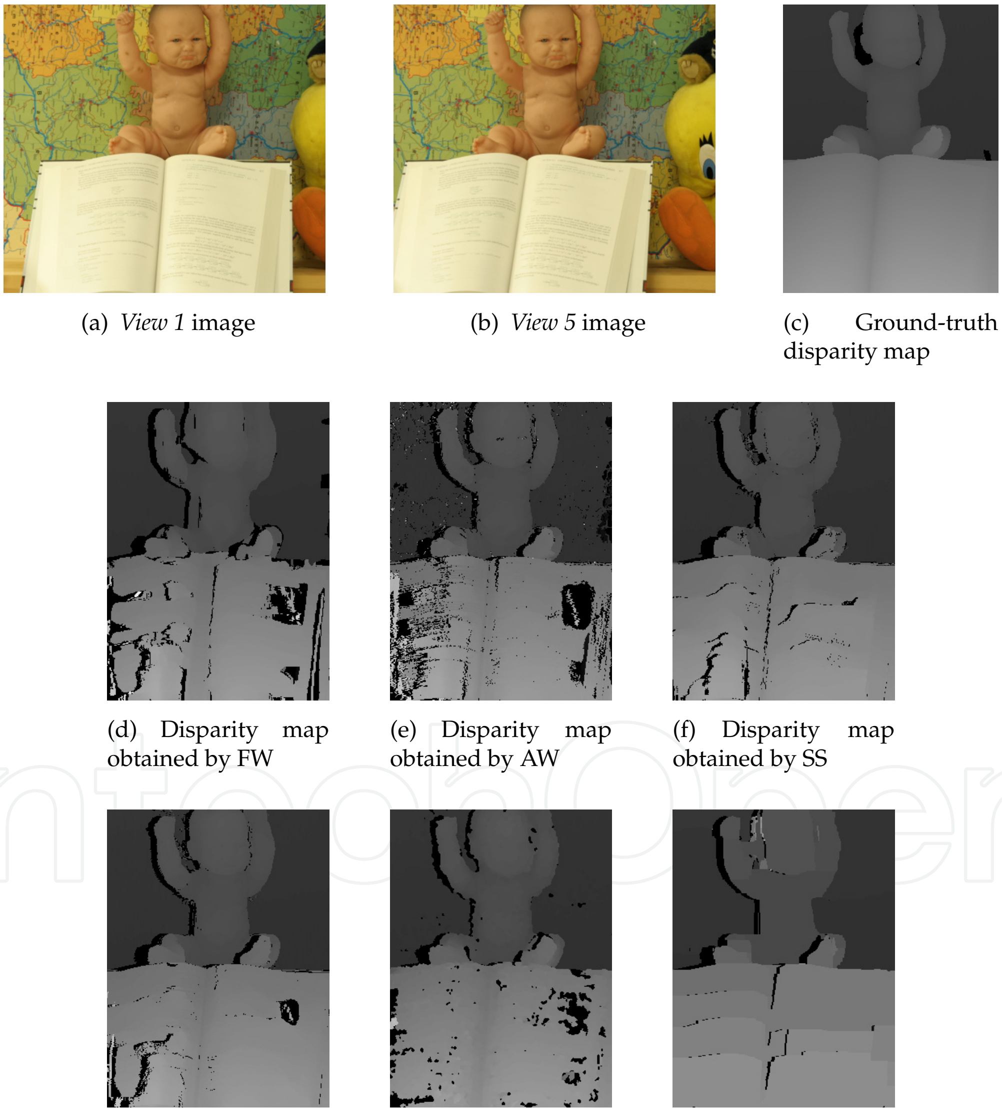
Figure 3 shows the depth maps obtained by applying the different stereo vision techniques on the *Baby2* scene from the Middlebury dataset along with the ground truth disparity provided by the website. The different techniques have different performance, but note how all of them introduce artifacts that will inevitably affect the final segmentation. Through this chapter, occluded points recognized by the stereo vision algorithms and not considered in the clustering and are reported in black in the pictures. Visible points (feature vectors) are, instead, reported with the color (different from black) of the cluster they belong to. That is, each cluster is associated to a color and pixels corresponding to points assigned by the segmentation algorithm to the same cluster share the same cluster color.

Figure 4 shows the results on the *Baby 2* image. The different rows correspond to the different segmentation algorithms while the columns correspond to the various stereo vision algorithms. All the proposed stereo vision and clustering algorithms work quite well on this scene and there is a sensible improvement from the usage of color or depth information alone (e.g., in the identification of the baby's feet). However the FW and GC algorithms produce some artifacts (especially close to the arm), that are then visible also in the segmentations. Also the number of points without a valid depth value is different for the various algorithms (FW and AW have larger missing areas). Probably on this scene the best performing algorithm is SGM. For what concerns the clustering techniques, differences are limited to some minor details.

Figures 6 refers instead to the *Aloe* image segmentation. All the employed algorithms are able to correctly recognize the plant and the vase by exploiting the joint usage of color and depth information. However FBS and SS provide slightly better performance while the FW algorithms has some problems in estimating the depth of this scene. Note also how the spectral clustering technique allows to avoid some artifacts that are present when using simpler clustering algorithms (specially with K-means).

Finally Figures 7 and 8 show an example of the proposed method outcome with data acquired by the LTTM laboratory setup as input. This scene has been used to evaluate the performance on a more realistic scene that has not been built for the purpose of stereo vision evaluation only. This scene present challenging regions for stereo vision algorithms. The structure of the scene is quite simple but the complex texture of the background can represent an issue for color-based segmentation methods. The performance is quite good, in most of the cases the basic structure of the scene is recognized and the shape of the person is correctly identified. However the artifacts of some of the stereo algorithms affect the boundary of the person shape. Furthermore while mean-shift and spectral clustering properly recognize the main objects by using k-means clustering the person is split into two clusters with a quite arbitrary boundary between them.

The superiority of the results based on both color and geometry versus the ones obtainable by just color or geometry is evident. The current section ends by evaluating the most effective stereo vision and clustering algorithms pair. Such an evaluation was performed on the basis of



(a) *View 1* image

(b) *View 5* image

(c) Ground-truth disparity map

(d) Disparity map obtained by FW

(e) Disparity map obtained by AW

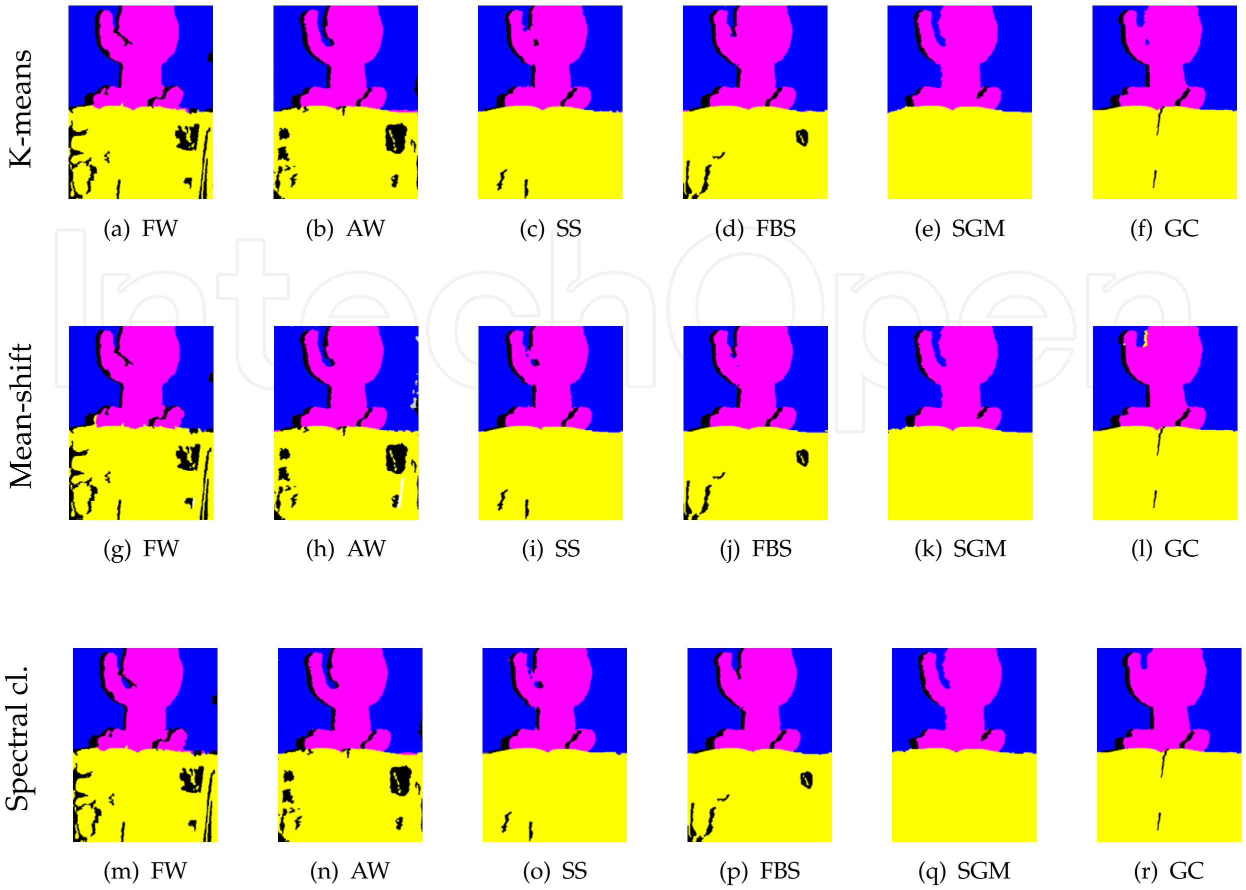
(f) Disparity map obtained by SS

(g) Disparity map obtained by FBS

(h) Disparity map obtained by SGM

(i) Disparity map obtained by GC

Figure 3. Middlebury “Baby 2” dataset stereo reconstruction results varying stereo algorithm.



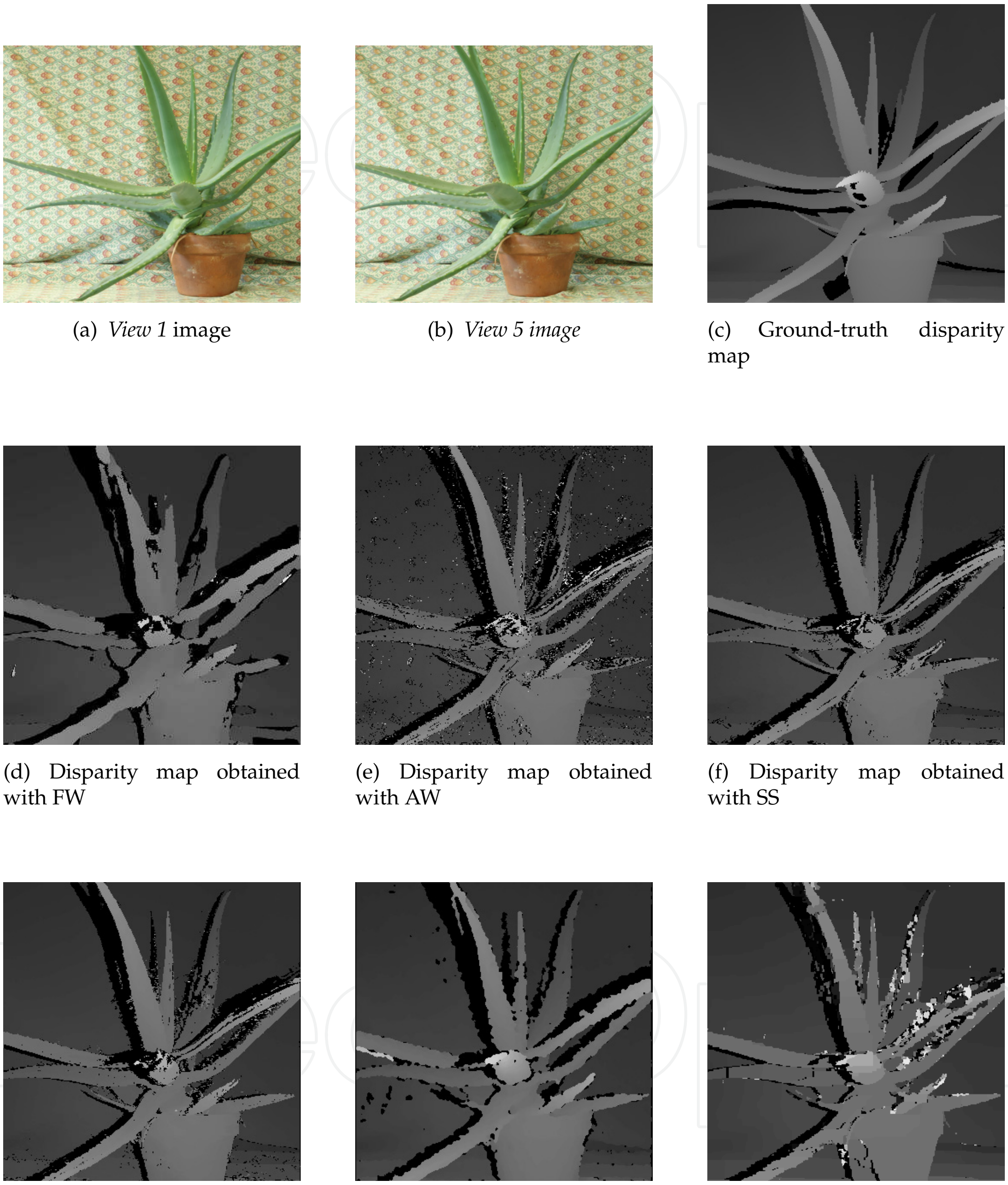
(a) FW (b) AW (c) SS (d) FBS (e) SGM (f) GC

(g) FW (h) AW (i) SS (j) FBS (k) SGM (l) GC

(m) FW (n) AW (o) SS (p) FBS (q) SGM (r) GC

Figure 4. Middlebury "Baby 2" dataset segmentation results varying stereo reconstruction algorithm.

a supervised metric computing the percentage of misclassified pixels with respect to a ground truth segmentation, obtained from the ground truth depth map provided for each scene of the Middlebury data-set. Occluded pixels were not taken into account during the computation. The percentages of misclassified pixels for all the eighteen combinations of stereo vision algorithms and clustering methods are reported in Table 2, together with the execution time of the stereo algorithms. Almost all the scene segmentations, with the exception of the ones obtained by applying k-means on the *person* scene are robust and effective, far way better than what is delivered by classical scene segmentation algorithms based on color information only. According to the considered metric, the most effective combination is given by SS stereo vision and spectral clustering with Nyström method. Unfortunately the SS algorithm is very slow. However, it is worth to note that the FSD algorithm [8] could be used in place of SS to obtain equivalent results much more quickly. It is also interesting to notice that the usage of global or semi-global stereo algorithms compared to the usage of local stereo algorithms does not lead to significant performance improvements. For example GC-based segmentation, especially combined with mean-shift clustering, on the person image (Figure 8) leads to more artifacts than local stereo vision algorithm. FBS appears to be a very good trade off between computational efficiency and segmentation precision and robustness, even if sometimes it may introduce false occlusions, as shown in Figure 4. K-means clustering does not work properly in the *person* scene (Figure 8). The more reliable clustering algorithm is spectral clustering with Nyström method, because it works robustly in all the scenes and with all the



(a) *View 1* image

(b) *View 5 image*

(c) Ground-truth disparity map

(d) Disparity map obtained with FW

(e) Disparity map obtained with AW

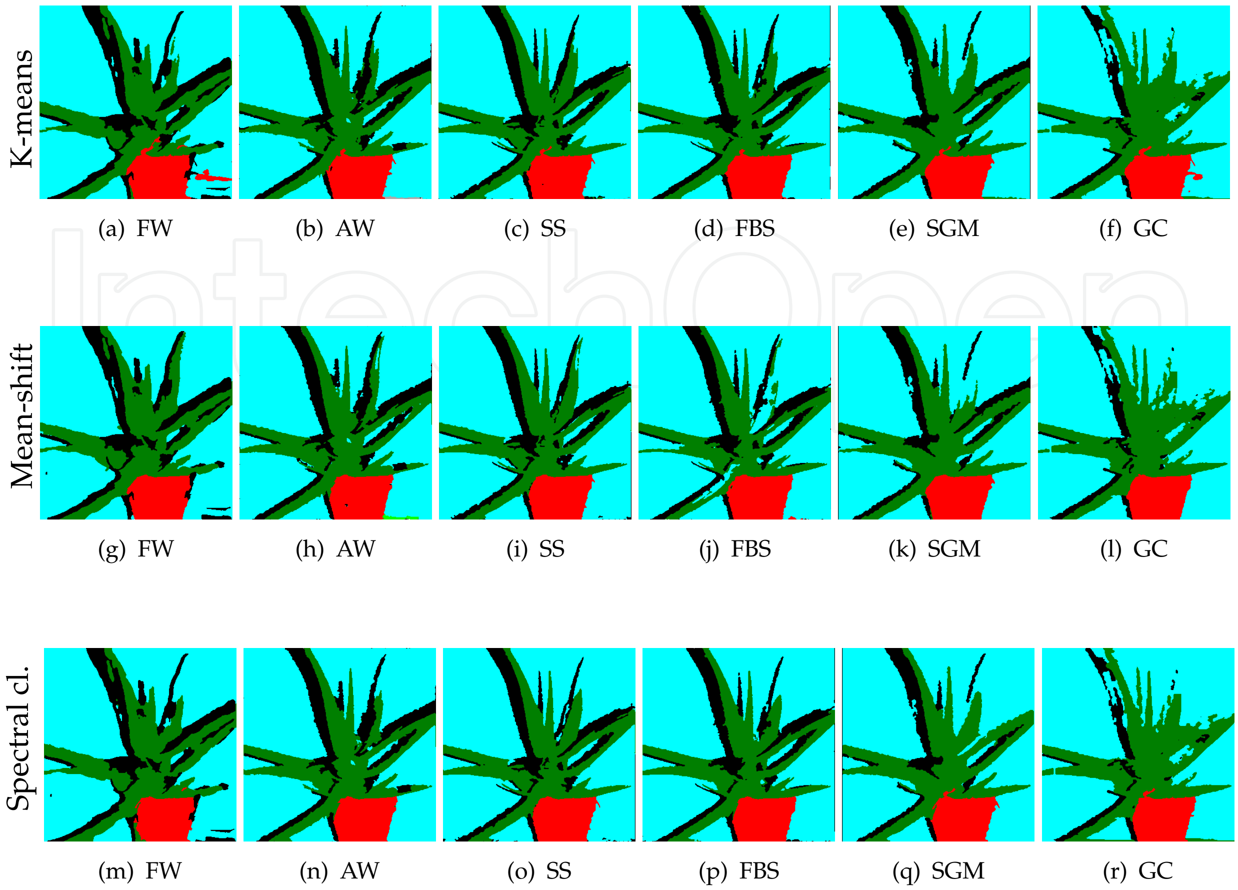
(f) Disparity map obtained with SS

(g) Disparity map obtained with FBS

(h) Disparity map obtained with SGS

(i) Disparity map obtained with GC

Figure 5. Middlebury "Aloe" dataset: disparity maps computed with different stereo vision algorithms.



(a) FW (b) AW (c) SS (d) FBS (e) SGM (f) GC

(g) FW (h) AW (i) SS (j) FBS (k) SGM (l) GC

(m) FW (n) AW (o) SS (p) FBS (q) SGM (r) GC

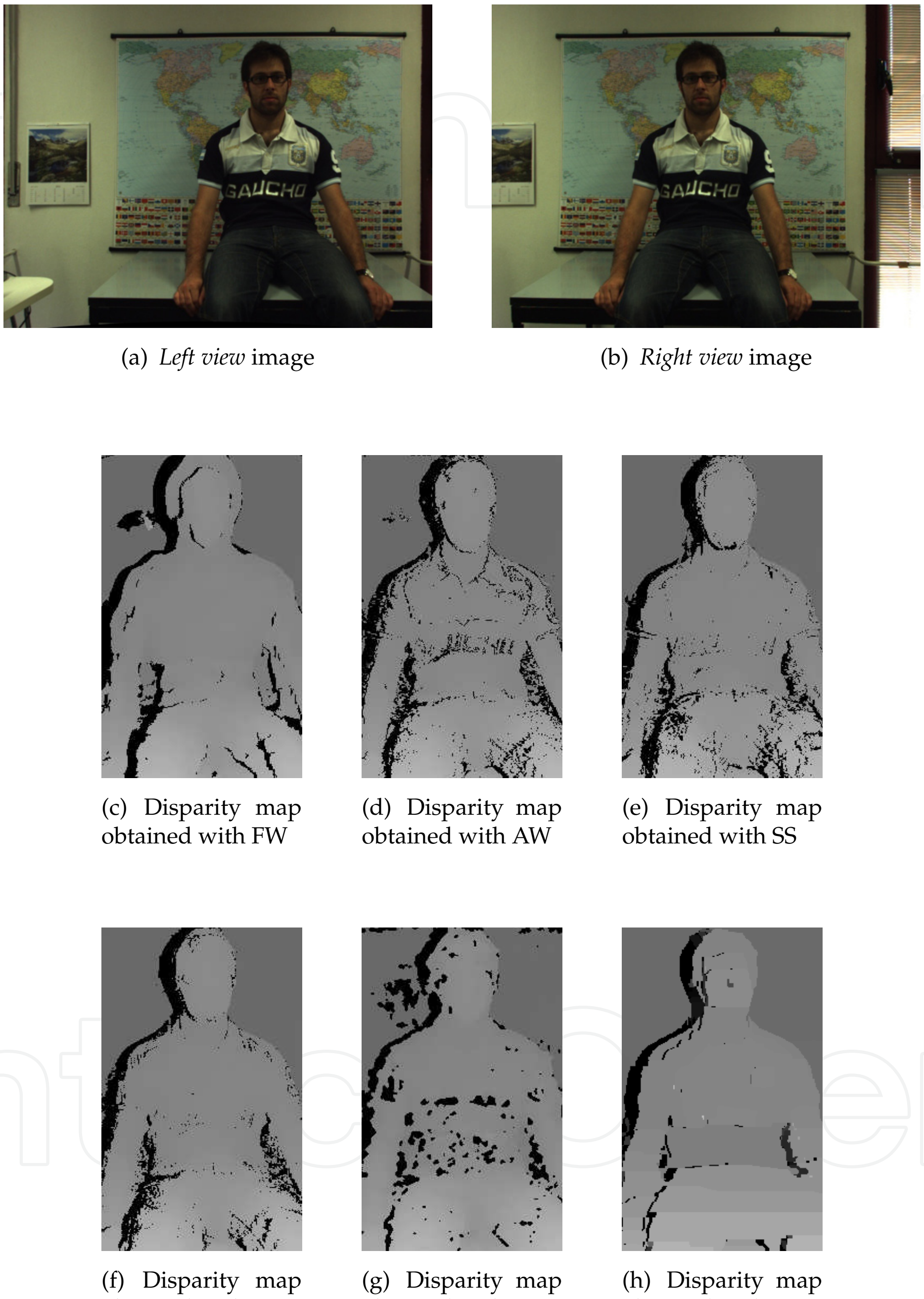
Figure 6. Middlebury "Aloe" dataset segmentation results with different stereo vision and clustering techniques.

stereo vision algorithms. In terms of speed, mean-shift clustering is slightly faster than the other two algorithms (that are comparable). All the Matlab implementations of the clustering algorithms take less than 7 seconds, allowing further real time applications with optimized implementations.

	FW	**AW**	**SS**	**FBS**	**SGM**	**GC**
k-means	2.21	0.87	0.92	0.92	1.60	0.97
Mean-shift	2.31	1.33	1.02	0.98	1.62	1.02
Spectral Clusteting	2.03	0.84	0.81	0.93	1.45	0.97

Table 2. Comparison with the segmentation performed on the Middlebury *baby 2* ground truth: percentage of incorrectly assigned pixels. The execution time (in [s]) is relative to the stereo algorithms only executed on a single core 2.53 GHz machine. GC and SGM are highly optimized algorithms, GC does not have subpixel refinement.

Finally note the importance of a correct setting of the λ parameter for the sake of an effective segmentation. Figure 9 depicts segmentation outcomes for the Segment Support algorithm by varying λ. Note how low and high values of λ lead to the undesired artifacts described in the chapter introduction and exemplified in Figures 1(b) and 1(c) respectively. In particular, the high values of λ give more importance to the estimated geometry, while lower values of λ give more importance to the color.



(a) *Left view* image

(b) *Right view* image

(c) Disparity map obtained with FW

(d) Disparity map obtained with AW

(e) Disparity map obtained with SS

(f) Disparity map obtained with FBS

(g) Disparity map obtained with SGM

(h) Disparity map obtained with GC

Figure 7. “LTTM Person" dataset stereo reconstruction results with different stereo vision algorithms.

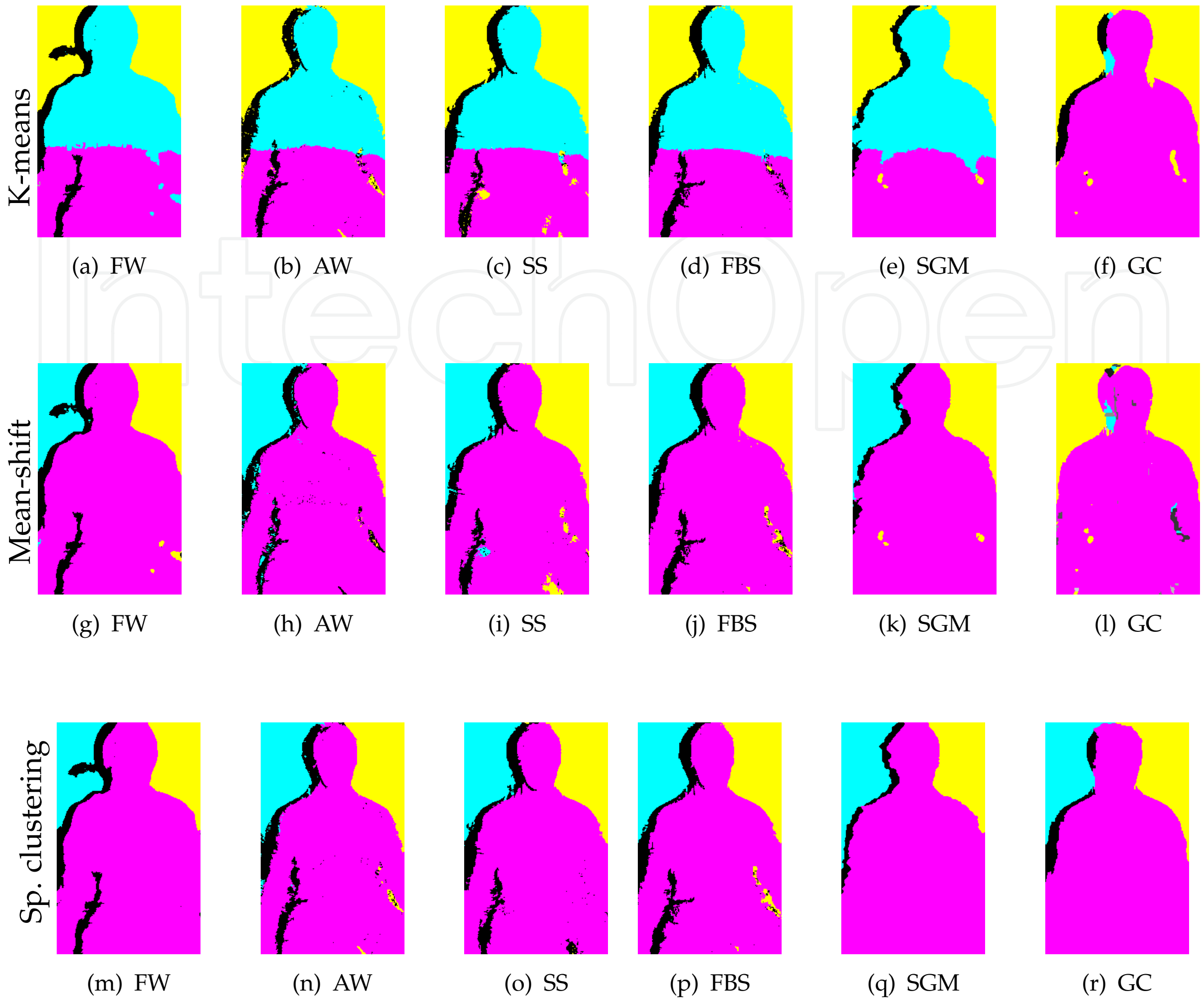


Figure 8. "LTTM Person" dataset segmentation results with different stereo vision and clustering techniques.

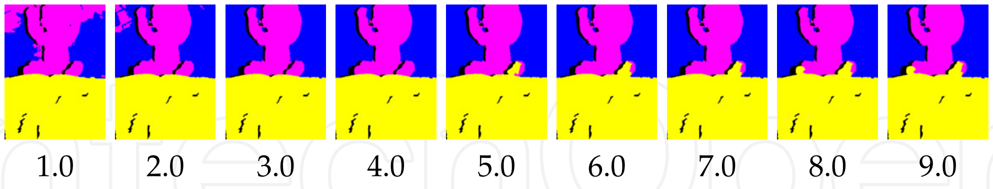


Figure 9. Segmentation results on the Middlebury Baby 2 scene corresponding to different values of the parameter λ (SS stereo vision algorithm).

5. Conclusions

This chapter shows how it is possible to synergically combine geometry and color information in order to obtain high quality scene segmentation. The geometry information, in particular, is obtained from stereo vision. Stereo vision techniques, historically employed to just extract 3D geometry from a pair of views of the framed scene, are therefore considered as a starting step for a segmentation pipeline where segmentation is eased and its efficiency improved by an enriched scene information that allows to solve most ambiguities that classical methods exploiting just color or geometry information are not able to solve.

Moreover, the experimental results show that the proposed approach can provide a better segmentation than the methods based on just color or just geometry. Since the main ingredients of the proposed approach are specific stereo vision and clustering algorithms, this chapter examines the results of the proposed approach with the different combinations of six different stereo vision and three different clustering algorithms. Among the various solutions the SS stereo vision algorithm combined with spectral clustering with Nyström method provides the best performance. Although this configuration is quite expensive in terms of execution time, the SS algorithm could be replaced by the much faster FSD algorithm to obtain equivalent results in a fraction of the time required by SS. The acquisition system needed for the proposed scene segmentation approach is a regular stereo vision system, essentially requiring two cameras instead of a single camera, as the standard color based segmentation methods. It is certainly true that two cameras form a more complex set-up than a single camera, but new applications are making increasingly common 3D acquisition systems, among which stereo vision ones are the most inexpensive and popular. The overall quality of the obtained results is good enough to justify such a modest complication of the acquisition system.

Future research may be devoted to the exploitation of the proposed scheme into stereo vision methods based on segmentation in order to improve both the segmentation and the quality of the extracted depth data, thus introducing an interesting coupling between the two problems.

Optimization of stereo vision algorithms for the segmentation task is an open field worthy to be explored. Newly developed depth cameras, like Time-Of-Flight cameras and structured light cameras (e.g., Microsoft Kinect) are a valid alternative for scene geometry estimation and the usage of different acquisition methods for 3D geometry in place of stereo reconstruction by color cameras will be taken into account.

Author details

Carlo Dal Mutto, Fabio Dominio and Pietro Zanuttigh
University of Padova, Italy

Stefano Mattoccia
University of Bologna, Italy

6. References

[1] Boykov, Y. & Kolmogorov, V. [2001]. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 26: 359–374.

[2] Brown, M. Z., Burschka, D. & Hager, G. D. [2003]. Advances in computational stereo, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25: 993–1008.

[3] Comaniciu, D. & Meer, P. [2002]. Mean shift: a robust approach toward feature space analysis, *Pattern Analysis and Machine Intelligence, IEEE Trans. on* 24(5): 603 –619.

[4] Crow, F. C. [1984]. Summed-area tables for texture mapping, *SIGGRAPH Comput. Graph.*

[5] Felzenszwalb, P. & Huttenlocher, D. [2004]. Efficient graph-based image segmentation, *Int. J. Comput. Vision* 59(2): 167–181.

[6] Fowlkes, C., Belongie, S., Chung, F. & Malik, J. [2004]. Spectral grouping using the nystrï£¡m method, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 26: 2004.

[7] Hirschmuller, H. [2006]. Stereo vision in structured environments by consistent semi-global matching, *Proc. of CVPR* 2: 2386–2393.

[8] Mattoccia, S. & De-Maeztu, L. [2011]. A fast segmentation-driven algorithm for stereo correspondence, *International Conference on 3D (IC3D 2011)*, Liege, Belgium.

[9] Mattoccia, S., Giardino, S. & Gambini, A. [2009]. Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering, *ACCV*.

[10] McDonald, M. [1981]. Box-filtering techniques, *Computer Graphics and Image Processing* 17(1): 65–70.

[11] Middlebury [2012]. Middlebury stereo vision website, http://vision.middlebury.edu/stereo/.

[12] OpenCV [2012]. Opencv, http://opencv.willowgarage.com/wiki/.

[13] Scharstein, D. & Szeliski, R. [2001]. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *International Journal of Computer Vision* 47: 7–42.

[14] Shi, J. & Malik, J. [2000]. Normalized cuts and image segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence* .

[15] Szeliski, R. [2010]. *Computer Vision: Algorithms and Applications*, Springer, New York.

[16] Tombari, F., S. Mattoccia, S. & Di Stefano, L. [2007]. Segmentation-based adaptive support for accurate stereo correspondence, *Proc. of IEEE Pacific-Rim Symp. on Image and Video Tech. 2007*, Springer.

[17] Yoon, K.-J. & Kweon, I. S. [2006]. Adaptive support-weight approach for correspondence search, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28: 650–656.