Joint 3D-Reconstruction and Background Separation in Multiple Views using Graph Cuts

Bastian Goldlücke and Marcus A. Magnor
Graphics-Optics-Vision
Max-Planck-Institut für Informatik, Saarbrücken, Germany
{bg,magnor}@grovis.de

Abstract

This paper deals with simultaneous depth map estimation and background separation in a multi-view setting with several fixed calibrated cameras, two problems which have previously been addressed separately. We demonstrate that their strong interdependency can be exploited elegantly by minimizing a discrete energy functional which evaluates both properties at the same time. Our algorithm is derived from the powerful "Multi-Camera Scene Reconstruction via Graph Cuts" algorithm recently presented by Kolmogorov and Zabih. Experiments with both real-world as well as synthetic scenes demonstrate that the presented combined approach yields even more correct depth estimates. In particular, the additional information gained by taking background into account increases considerably the algorithm's robustness against noise.

1. Introduction

The reconstruction of the 3D-geometry of a scene from multiple views remains a challenging problem in computer vision. It has been approached from several points of view, one of which is energy minimization. Until recently, it was not feasible to make use of its theoretical advantages in practice, since minimizing an energy functional is usually NP-hard. Recently, however, algorithms have been developed to overcome this drawback. One class of methods formulates the problem in terms of level sets and uses numerical techniques to obtain a local minimum as the solution of a set of PDEs [1]. We follow a different approach, in which the energy functional is discrete, and graph cuts are employed iteratively to successively reduce the energy. Many vision problems have been treated successfully with this kind of energy minimization, including stereo and motion [2, 3, 5, 9] and voxel occupancy [12]. Evaluations of stereo algorithms using real images for which true dense depth information is known indicate that minimization algorithms based on graph cuts yield very good results [10]. Kolmogorov and Zabih introduced a fairly general class of energy functionals for which they proved that it is possible to minimize them via graph cuts [7]. They used this mathematical framework in [6] to construct an algorithm for multi-camera scene reconstruction which performs very well with real imagery.

On the other hand, the separation of the foreground of a scene from a known background is another important prerequisite for several interesting vision algorithms. In particular, the computation of the visual hull relies entirely on object silhouette information.The kind of separation we have in mind is most closely related to video matting techniques, several of which are widely used. The *blue screen method* and *multi-background matting* rely on backgrounds with special mathematical properties and require a tightly controlled studio environment to be successful [8]. Our approach falls into a second category which uses *clean plates*, images of the static background of the scene.

Clearly, 3D-reconstruction as well as background separation could benefit greatly from a known solution to the respective other problem: If the static background pixels in an image are known, then these pixels must have the same depth as the background, while all other pixels must be less deep. On the other hand, if we know the depth of each pixel, then only pixels with a lesser depth than the background can belong to the foreground.

In the following sections we describe an algorithm which exploits this interdependency by addressing both problems simultaneously, assuming that we have a set of fully calibrated cameras and an image of the static background for each camera with at least approximate per-pixel depth information. We present a generalization of the successful multi-view reconstruction algorithm from [6]. Pixels are not only labeled by their depth, but also by an additional flag which indicates whether a pixel belongs to the background or not. As in the original method, the result of our depth reconstruction and background separation algorithm is obtained

as the minimum of an energy functional. Besides taking into account classical constraints from multi-view stereo, it regards the new considerations related to background as well.

Sect. 2 outlines the problem we want to solve precisely and introduces the notation which is used throughout the rest of the paper. The energy functional we minimize is defined in Sect. 3, while Sect. 4 is devoted to the method of graph cuts, which is used to perform this minimization. There we also sketch a proof that this method is applicable to our energy functional. Results we achieve by applying our algorithm to real-world as well as synthetic data are demonstrated in Sect. 5. Finally we conclude with a summary and some ideas for future work in Sect. 6.

2. Reconstruction Algorithm

We aim at reconstructing the 3D-geometry of a static scene captured by a number of calibrated cameras directly from the images. The goal is to retrieve depth maps, assigning a depth value to each pixel which defines its location in 3D-space. Simultaneously, we want to decide for every pixel whether it belongs to the background of the scene, known from *background images* captured with the same cameras. We assume that the depth of each pixel in the background images can be estimated at least approximately. In practice, we use our algorithm with background subtraction turned off to obtain this background configuration.

Our algorithm is a generalization of the *multi-camera scene reconstruction via graph cuts* [6]. It shares its advantages: All input images are treated symmetrically, visibility is handled properly, and spatial smoothness is imposed while discontinuity is preserved. While our energy functional is different, we utilize a similar problem formulation and notation, which we introduce now.

Input: The input to the algorithm is the set of pixels $\mathcal{P}_k$ from each source camera k together with the following mappings for every pixel $p \in \mathcal{P} := \bigcup_k \mathcal{P}_k$:

$I(p)$	The color value of the input image.
$\Delta I(p)$	The value of the (discretely evaluated) Laplacian of the input image.
$B(p)$	The color value of the background image.

Output: The goal is to find the "best" mapping $\lambda : \mathcal{P} \to \mathfrak{L}$ into a set of *labels* $\mathfrak{L}$. The precise definition of "best" is given later. To each pixel is assigned a label $\mathfrak{l} = (\mathfrak{l}_d, \mathfrak{l}_b)$, which is a *pair* of values. This is our first generalization: Labels not only encode depth, but also the new property of "backgroundness". The boolean value $\mathfrak{l}_b$ is true if and only if p is a background pixel, while $\mathfrak{l}_d$ denotes the *depth* of p.

As is done in the original algorithm [6], the notion of "depth" we use is a somewhat abstract one: Depth labels correspond to level sets of a function $D : \mathbb{R}^3 \longrightarrow \mathbb{R}$ satisfying for all scene points $P, Q \in \mathbb{R}^3$ and all cameras k:

$$P \text{ occludes } Q \text{ in } k \;\Rightarrow\; D(P) < D(Q).$$

This is obviously a very natural requirement for a function indicating depth. The existence of such a function D implies that there is a way to define depth *globally*, i.e. independent of a specific camera. The same constraint is postulated in the original algorithm [6] as well as in voxel coloring [11]. An important special case in which the constraint is automatically satisfied occurs when all cameras are located on one side of a plane P looking at the other side. The level sets of D can then be chosen as planes which lie parallel to P.

Topology: The definition of the algorithm includes the topological properties of the input images. A set-theoretic description is given by assigning to every $p \in \mathcal{P}$ the following sets of pixels:

$\mathcal{N}_p$	A set of neighbors of p in $\mathcal{P}_k$ *excluding* p where the energy functional will encourage continuity.
$\mathcal{C}_p$	A neighborhood of p *including* p. These regions will later be relevant for the computation of normalized cross correlations which are used as a criterion for photo-consistency.

Geometry: Finally, the geometric relations between pixels in different images with regard to their current labels and the camera positions must be specified. We encode these in the set $\mathfrak{I}$ of *interactions*. First note that a pixel p together with a label $\mathfrak{l}$ corresponds to a point in 3D-space via the projection parameters of the camera. This point is denoted by $\langle p, \mathfrak{l} \rangle$. The interactions now represent a notion of "nearness" of two 3D-points in the following sense, Fig. 1:

- A pair $\{\langle p, \mathfrak{l} \rangle, \langle q, \mathfrak{l} \rangle\}$ belongs to $\mathfrak{I}$ if and only if
  1. $q \in \mathcal{P}_k$ and $p \notin \mathcal{P}_k$, i.e. p and q must come from two different cameras.
  2. q is the pixel nearest to the projection of $\langle p, \mathfrak{l} \rangle$ onto the image of camera k.

  Note that interacting pixels always have the same label.

The set $\mathfrak{O}$ of *occlusions* will be used to enforce visibility constraints. It also contains pairs of 3D-points and is defined as follows:

- A pair $\{\langle p, \mathfrak{l} \rangle, \langle q, \mathfrak{l}' \rangle\}$ belongs to $\mathfrak{O}$ if and only if $\{\langle p, \mathfrak{l} \rangle, \langle q, \mathfrak{l} \rangle\} \in \mathfrak{I}$ and $\mathfrak{l}_d < \mathfrak{l}'_d$. Geometrically this means that if $\langle p, \mathfrak{l} \rangle$ is projected onto q, then it will occlude q if and only if the depth assigned to p is smaller than the depth assigned to q.

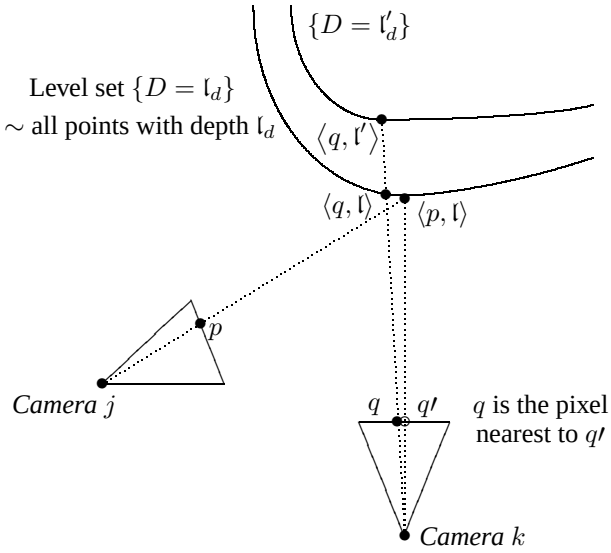

Figure 1. Interactions and Occlusions. The 3D-points $\langle p, \mathfrak{l} \rangle$ and $\langle q, \mathfrak{l} \rangle$ interact, thus $\{\langle p, \mathfrak{l} \rangle, \langle q, \mathfrak{l} \rangle\} \in \mathfrak{I}$. On the other hand, $\langle q, \mathfrak{l}' \rangle$ is occluded by $\langle p, \mathfrak{l} \rangle$ in its camera image, so $\{\langle p, \mathfrak{l} \rangle, \langle q, \mathfrak{l}' \rangle\} \in \mathfrak{O}$.

Energy minimization: The best configuration λ corresponds to the one that minimizes an *energy functional* $E(\lambda)$. This functional encodes the high level knowledge about scene reconstruction: Unlikely or impossible assignments of labels must be penalized, while very likely configurations must be enforced. A precise definition of the energy functional we use is given in the next section.

3. The Energy Functional

The energy functional which is minimized by the algorithm can be written as a sum of contributions by every single pixel and every possible pair of pixels:

$$E(\lambda) = \sum_{p,q \in \mathcal{P}} \left[E^{p,q}_{\text{photo}}(\lambda) + \beta E^{p,q}_{\text{smooth}}(\lambda) + E^{p,q}_{\text{vis}}(\lambda) \right] + \alpha \sum_{p \in \mathcal{P}} E^{p}_{\text{background}}(\lambda).$$

The terms on the right hand side will be different from zero only if p and q interact or occlude each other in certain configurations, or if p and q are neighbours. Thus, the sum runs in effect only over relatively few pairs of points. The positive weights α and β are the only free parameters of our method. Good choices will be specified in Sect. 5. The goal of the graph cut algorithm in Sect. 4 is to find an assignment λ of labels to all pixels that is a local minimum of E in a strong sense.

We now give a detailed description of the four contributing terms.

3.1. Photo-consistency term

For interacting pixels sharing similar characteristics, we issue a photo-consistency bonus. This reflects the fact that if a 3D-point is projected onto a pixel p in one image and a pixel q in another and is visible in both images, then pixels in the neighbourhoods $\mathcal{C}_p$ and $\mathcal{C}_q$ should be similar. Mathematically, we set

$$E^{p,q}_{\text{photo}}(\lambda) := \begin{cases} -C(p,q) & \text{if } \{\langle p, \lambda(p) \rangle, \langle q, \lambda(q) \rangle\} \in \mathfrak{I}, \\ 0 & \text{otherwise.} \end{cases}$$

The *correlation term* $C(p,q) \in [0,1]$ must be small if $\mathcal{C}_p$ differs from $\mathcal{C}_q$ and large if the local pixel neighbourhoods are very similar. We found experimentally that a very good criterion is the statistical measure obtained by computing

- The normalized cross-correlation[1] between the sets of color values $I(\mathcal{C}_p)$ and $I(\mathcal{C}_q)$, taking the minimal correlation among the three color channels, and
- The normalized cross-correlation between the sets of Laplacians $\Delta I(\mathcal{C}_p)$ and $\Delta I(\mathcal{C}_q)$, again computing the three color channels and taking the minimum.

A weighted average of these two values is then assigned to $C(p,q)$. In both cases the neighborhoods we use are square 3×3 pixel windows surrounding the points.

Indeed, this scheme has theoretical advantages as well. Especially in real-world data, correlations are much more robust than some kind of distance measure between the color values: Stereo images taken simultaneously by different cameras often have significantly different color values even for corresponding pixels, because the response of the cameras to the same signal is not identical. This effect can be somewhat reduced by careful calibration, but it remains a principal problem. Since correlation measures statistical similarity, not absolute similarity in values, it yields more reliable results even with uncalibrated images. This is especially true for neighbourhoods containing edges, which are generally more easily matched.

To further encourage that image features like edges and corners are matched with their counterparts in other images, we include the correlation of the Laplacian of the image into $C(p,q)$.

3.2. Smoothness term

Drastic changes in depth or transitions from background to foreground are usually accompanied by image features.

[1] Cross-correlations in our sense are always positive numbers. If the result from the computation is negative, it is set to zero.

We transfer this simple observation into the smoothness energy

$$E^{p,q}_{\text{smooth}}(\lambda) \;:=\; V^{p,q}\big(\lambda(p),\lambda(q)\big), \text{ where } V^{p,q}(\mathfrak{l},\mathfrak{l}') \;:=$$
$$:= \quad \begin{cases} 0 & \text{if } q \notin \mathcal{N}_p \text{ or } \mathfrak{l} = \mathfrak{l}', \\ 2L_{\max} \;-\; \|\Delta I\,(p)\|_\infty \;-\; \|\Delta I\,(q)\|_\infty & \text{otherwise.} \end{cases}$$

If the pixels are neighbors, it penalizes changes in depth or "backgroundness" if image colors vary only slightly in the neighborhood of p or q. We enforce smoothness only in the four nearest neighbors, of which the set $\mathcal{N}_p$ consists in our case. The Laplacian of the image is used as a simple edge detector. The maximum norm in the above definition denotes the maximum of all color channels, so a change in any channel is sufficient for the presence of a feature, which is a natural assumption. $L_{\max}$ is the largest possible absolute value for the Laplacian, which depends on color encoding and level of discretization.

3.3. Visibility constraints

Certain configurations of labels are impossible because of occlusions. If camera j sees pixel p at depth $\mathfrak{l}_d$, and the projection of $\langle p, \mathfrak{l}\rangle$ into another image is pixel q, then it is of course not possible that q has a larger depth than p. These illegal configurations are precisely the ones captured by the set of occlusions, so we forbid them by assigning an infinite energy

$$E^{p,q}_{\text{vis}}(\lambda) \;:=\; \begin{cases} \infty & \text{if } \{\langle p,\lambda(p)\rangle\,,\langle q,\lambda(q)\rangle\} \in \mathfrak{O}, \\ 0 & \text{otherwise.} \end{cases}$$

3.4. Background term

For the classification of pixels as background pixels we again use normalized cross-correlations $C_b(p)$, this time computed between the ordered sets of image colors $I\,(\mathcal{N}_p)$ and background colors $B\,(\mathcal{N}_p)$. We penalize good correlations of the image values with the background values if λ does not classify p as a background pixel. A second constraint is the background depth: If $\lambda_b(p) = \text{true}$, i.e. p belongs to the background, then p must have the same depth $\mathfrak{b}_d(p)$ as the background. This results in the following formula:

$$E^{p}_{\text{background}}(\lambda) \;:=\; \begin{cases} C_b(p) & \text{if } \lambda(p)_b = \text{false}, \\ \infty & \text{if } \lambda(p)_b = \text{true} \\ & \text{and } \lambda(p)_d \neq \mathfrak{b}_d(p), \\ 0 & \text{otherwise.} \end{cases}$$

In image areas with few texture information, it is often the case that the correlation $C_b(p)$ is low even if p is really a background pixel. For this reason we do not penalize low correlations when the current labelling λ classifies p as background.

4. Energy Minimization

In this section we scetch a formal proof that graph cuts can be used to find a strong[2] local minimum of our energy functional. The algorithm works by iterating over all labels, deciding in each step which pixels have to be changed to the current label in order to reduce the energy. One can start with any valid configuration λ_0 with $E(\lambda_0) < \infty$. An obvious choice is to set each pixel to the maximum possible depth and tag it as foreground. Since the energy is always reduced and impossible configurations have infinite energy, only valid configurations can be generated. We will now investigate a single step of the iteration in more detail.

Let λ be the current label configuration of all pixels and $\mathfrak{a}$ the current label considered. Any set of pixels $\mathcal{A} \subset \mathcal{P}$ determines a new labelling $\lambda_{\mathcal{A},\mathfrak{a}}$ via an $\mathfrak{a}$-*expansion*: Set for every $p \in \mathcal{P}$

$$\lambda_{\mathcal{A},\mathfrak{a}}(p) \;:=\; \begin{cases} \mathfrak{a} & \text{if } p \in \mathcal{A}, \\ \lambda(p) & \text{otherwise.} \end{cases}$$

The goal of each step is to determine $\mathcal{A}$, i.e. the set of pixels to be assigned label $\mathfrak{a}$, such that the energy becomes smaller if at all possible, otherwise it should stay the same – formally we want $E(\lambda_{\mathcal{A},\mathfrak{a}}) \leq E(\lambda)$. A very efficient algorithm achieving this uses graph cuts [7]. We do not repeat this construction here and only prove that it can be applied to our case.

First the energy funcional must be rewritten in a way which captures *energy changes* during the possible $\mathfrak{a}$-expansions. Therefore we number the pixels in $\mathcal{P}$,

$$\mathcal{P} =: \{p_1, \ldots, p_N\},$$

and define for each $i = 1, \ldots, N$ a function of a binary variable

$$\sigma_i : \{0,1\} \to \mathfrak{L}, \quad \sigma_i(x) \;:=\; \begin{cases} \mathfrak{a} & \text{if } x = 1 \\ \lambda(p_i) & \text{otherwise.} \end{cases}$$

We can now define an energy $E_{\lambda,\mathfrak{a}}$ depending on N binary variables which encode whether the label of the corresponding pixel is changed during the $\mathfrak{a}$-expansion or not:

$$E_{\lambda,\mathfrak{a}} : \{0,1\}^N \to \mathbb{R},$$
$$E_{\lambda,\mathfrak{a}}(x) := E\big(\sigma_1(x_1), \ldots, \sigma_N(x_N)\big).$$

The task of finding the set $\mathcal{A}$ is then equivalent to the task of finding a vector $x \in \{0,1\}^N$.

In consideration of Theorem 3 in [7], it is sufficient to prove the following lemma for the energy functional E defined in the last section.

[2] "strong" in the same sense as in [3]

Lemma. *Determine functions E^i and $E^{i,j}$ of one or two binary variables, respectively, such that for all $x \in \{0,1\}^N$*

$$E_{\lambda,\mathfrak{a}}(x) \;=\; \sum_{1\le i\le N} E^i(x_i) \;+\; \sum_{1\le i<j\le N} E^{i,j}(x_i,x_j).$$

Then each term $E^{i,j}$ satisfies the condition

$$E^{i,j}(0,0) + E^{i,j}(1,1) \;\le\; E^{i,j}(0,1) + E^{i,j}(1,0).$$

Proof. In view of the arguments in [6, Sect. 4.2], the only thing we have to show is that V^{p_i,p_j} is a metric. A detailed proof can be found in the extended online version of the paper on our web page [13]. □

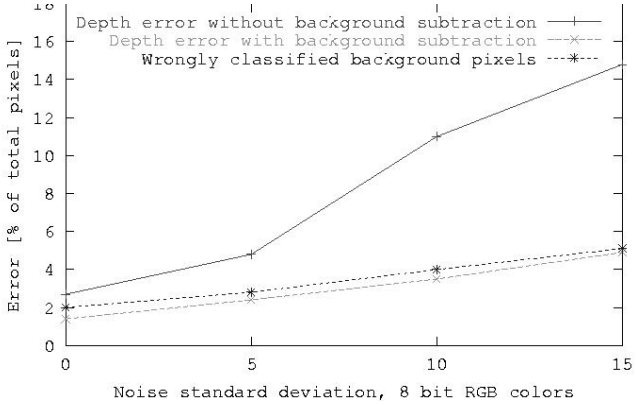

Figure 2. Dependence of the depth estimation and background separation error on the amount of noise added to the image in Fig. 4.

5. Results

We first test the quality of the depth maps computed by our method in conjunction with our real-time dynamic light field rendering application [4]. The system is capable of rendering scenes from novel viewpoints inside the window spawned by the cameras. The quality of the rendering mainly depends on good per-pixel depth information. We use data from the Stanford Multi Camera Array, a 3×2 array of CMOS imagers with parallel optical axes. The cameras are relatively far apart in our examples, which makes 3D-reconstruction more difficult due to the large disparity range from 3 to 34 pixels at an image resolution of 320×240 pixels. There are also dissimilarities in the color reproduction of the cameras, as well as artifacts due to MPEG compression during acquisition, imposing a further challenge onto color matching.

Fig. 3 depicts a frame of the sequence and the static background from one camera as well as the results from depth estimation and background separation. We extended our original rendering algorithm to make use of the additional background separation. It now renders first the constant background from the novel viewpoint, and then splats the foreground onto it. This method results in sharper edges and little bluriness in the final result. The overall sharpness in our rendering results indicates that the depth maps are in most areas very accurate, since each pixel is the result of a blending scheme where the two source images are weighted equally.

For a formal verification of our method, we render a complex synthetic scene from four different viewpoints and use the Z-Buffer to obtain true per-pixel depth information. We run our algorithm to reconstruct depth and background information and compare the outcome with the known ground truth. Fig. 4 shows an image of the scene and some of the results. The reconstruction error is defined as the percentage of pixels for which a depth value is computed that is off by more than one pixel in disparity. Results from the new algorithm with background separation are compared to results with background separation turned off in order to demonstrate the benefits of our method in comparison to [6], Fig. 2. In the case with background separation, the percentage of pixels which are wrongly classified as background or foreground is also determined.

To verify the robustness of our algorithm, we perturb the color values of the input images with a preset amount of noise. To each color channel in each pixel we add a random number from a Gaussian distribution with mean zero and standard deviation σ. Here the true strength of our algorithm becomes evident. The residual error is already almost halved when compared to the original algorithm in the noiseless case, but the results of our new method remain well below 5% error even when a significant amount of noise is introduced. For the final case of $\sigma = 15$ grey levels, the results from the algorithm without background separation are almost useless, while our algorithm quite robustly gives only 4.9% faulty assigned pixels.

Both methods are running using optimal parameters, which are found to be the same in both cases - we experimentally determined $\alpha = 0.6$ and $\beta = 0.4$. Fig. 4 displays the result of our reconstruction with a noise standard deviation of $\sigma = 5$. Disparity values range from 2 to 20 pixels.

After 30 seconds of one-time initialization to precompute all correlations, one full cycle of iterations over all labels takes 65 seconds on a 1.8GHz Pentium III Xeon. We found that usually about four cycles are needed for convergence to the final result, so it takes a total amount of 290 seconds to compute all data for four 320×240 pixel images.

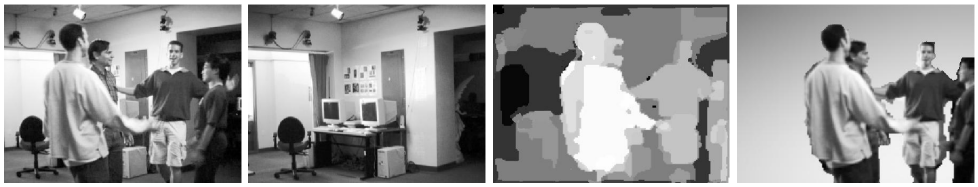

Figure 3. From left to right: (i) Scene image, (ii) background image, (iii) reconstructed depth labels and (iv) the detected foreground.

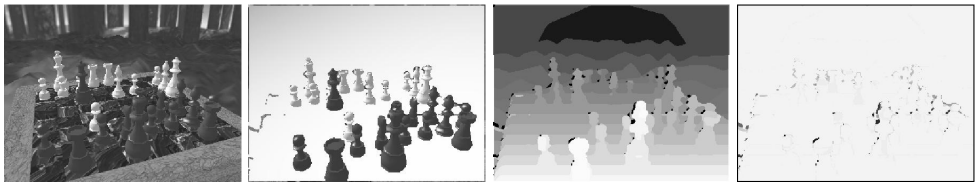

Figure 4. From left to right: (i) Synthetic scene, (ii) result of the background subtraction, (iii) reconstructed depth labels and (iv) the distribution of the residual depth error compared to the known ground truth. The amount of Gaussian noise added is set to $\sigma = 5$ grey levels.

6. Summary and Conclusions

We have presented a homogenous approach to simultaneous 3D-reconstruction and background separation from multiple views of the same scene. Our results clearly demonstrate that a joint solution benefits both problems: The continous background feedback from the current estimate improves the reconstruction and vice versa.

Moreover, it is a natural generalization of an already very successful reconstruction method based on minimizing a discrete energy functional via graph cuts. Existing code can be easily adapted to include background separation. Additionally, we provide a Linux implementation of the method and all necessary data to reproduce the examples on our web page [13].

Since the algorithm is extremely flexible, it should be possible to incorporate even more visual clues into its unifying framework. Our future work will include investigating how to exploit temporal coherence in video streams to further improve the reconstruction, as well as work on lifting the current constraints on camera geometry.

References

[1] L. Alvarez, R. Deriche, J. Sánchez, and J. Weickert. Dense disparity map estimation respecting image discontinuities: A PDE and scale-space based approach. *INRIA Rapport de Recherche, No.3874*, Jan. 2000.

[2] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *International Conference on Computer Vision*, pages 489–495, 1999.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, Nov. 2001.

[4] B. Goldlücke and M. Magnor. Hardware-accelerated dynamic light field rendering. In *Vision, Modeling and Visualisation*, 2002.

[5] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *European Conference on Computer Vision*, pages 232–248, 1998.

[6] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV (3)*, pages 82–96, 2002.

[7] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *ECCV (3)*, pages 65–81, 2002.

[8] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan. Image-based 3D photography using opacity hulls. In *Proceedings of ACM SIGGRAPH*, pages 427–436, 2002.

[9] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *International Conference on Computer Vision*, 1998.

[10] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3), April-June 2002.

[11] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *CVPR*, pages 1067–1073, 1997.

[12] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 345–352, 2000.

[13] *www.mpi-sb.mpg.de/∼bg/depth.html*.