Fast variational multi-view segmentation through backprojection of spatial constraints

C. Reinbacher *, M. Rüther, H. Bischof

Institute for Computer Graphics and Vision, Graz University of Technology, Inffeldgasse 16/III, Graz, Austria

Abstract

We present a variational segmentation method which exploits color, edge and spatial information between an arbitrary number of views. In contrast to purely image based information like color and gradient, spatial consistency is a new cue for segmentation, which originates from the field of 3D reconstruction. We show that this cue can be easily integrated in a variational formulation and allows pixel-accurate segmentation, even for objects which are hard to segment. The use of inherently parallel algorithms and the implementation on modern GPUs allows us to apply this method to semi-supervised and completely automatic settings. On publicly available datasets we show that our method is faster and more accurate than the state of the art. The successful applications within a catadioptric measurement system and multi-view background subtraction shows its practical relevance.

1. Introduction

Identifying the outline of an object is a crucial first step in applications like image based reconstruction, object tracking and pose estimation.

Most automatic segmentation approaches rely on a static and known background, whereas in semi-supervised segmentation the user guides the process by providing foreground and background seeds. Most methods operate on single images, even if several views of the same object are available. In this paper we present a segmentation method which fuses foreground and background information over multiple calibrated views in a geometrically consistent way. This spatial information greatly enhances the segmentation cues compared to the ones obtained from single images only (e.g. Fig. 1).

1.1. Previous work

Simultaneous segmentation of an object over different views is a problem recently addressed as image cosegmentation [1–4]. While cosegmentation is not clearly defined, it can be described as segmenting similarly looking objects. The field can be divided into two scenarios: a) rough segmentation of an object class e.g.: cars in roadside scenarios, or b) accurate segmentation of a particular object captured from different viewpoints. Most cosegmentation methods tackle the first problem (e.g. [2,4]), while recently progress has been made in the latter case [3]. In this paper we focus on the second case which poses hard challenges like changes in illumination and appearance of the object, caused by very large camera baselines. Vicente et al. [3] tackle this problem by learning from ground-truth segmentations of pairs of similar objects. If the camera positions are known (e.g. from Structure from Motion methods), this information can be used to fuse segmentation in different views in a spatially consistent way without requiring a geometric model of the object.

The principal assumption is, that foreground regions of a particular object in different views must originate from a projection of the same 3D space. This spatial coherence has been exploited in some early works [5,6] and more recently by Kolmogorov et al. [7]. The methods use depth information obtained from stereo images to preserve a foreground region under a changing background. They show a clear improvement compared to single image cues, but do not generalize to more than two views.

In a multi-view case it is advantageous to enforce spatial coherence with a common 3D region instead of pixel-wise depth information. The projections of this 3D region in all input images should completely

© 2012 Elsevier B.V. All rights reserved.
overlap the foreground regions. The inverse problem, i.e. the estimation of the 3D region from 2D foreground regions is usually referred to as Visual Hull estimation [8]. One of the first methods using the principle of voxel occupancy was presented by Snow et al. [9]. Their method relied on background images to generate a reconstruct a spatially smooth object surface. Zeng and Quan [10] try to maximize the volume of the visual hull by enforcing photo consistency between the extracted foreground regions from different viewpoints. However, their formulation only approximates the spatial consistency, thus it deviates from the strict visual hull assumption. Yezzi and Soatto [11] presented a method which simultaneously estimates a smooth object surface and radiance. Due to the smoothness assumption in the object albedo, their method does not work in the presence of texture or sharp edges on the object. Other methods make use of a shape prior either by minimizing differences between silhouette regions in successive frames [12], or by using a shape model to guide the segmentation [13].

Recently, progress has been made by incorporating color and spatial coherence using segmentation in a regular discretization of the 3D space. Franco and Boyer [14] use a space occupancy grid to fuse 2D probability maps obtained from known background models into a probabilistic 3D occupancy grid. Campbell et al. [15] employ a globally optimal volumetric graph-cut to incorporate color models from several input images. Their method is fully automatic, only requiring the object to be in the center of every image. Kolev et al. [16,17] fuse constraints obtained from user input in a 3D voxel space and carry out a globally optimal segmentation directly in 3D space. In these methods, a coarse 2D segmentation is only a by-product, given by projecting the reconstructed 3D shape into the input images. As the voxel resolution usually is much lower than the pixel resolution, this does not lead to pixel-accurate segmentations and thus does not scale well to high resolution cameras. Lee et al. [18] circumvent this problem by formulating the spatial consistency in pixel space. However, their iterative silhouette estimation procedure relies on intermediate segmentation results which in turn may be erroneous.

1.2. Contribution

In this paper we present a novel segmentation method which makes use of the additional information contained in multiple calibrated views of an object. In contrast to existing methods, we are able to combine single view information (color and edge) with spatial information between different views in a consistent and accurate way. The segmentation is carried out by solving a globally optimal energy minimization problem in image space. Our method is inherently parallel, and we are able to process 8 cameras (two megapixels) in under a second. This allows us to apply the method to a variety of problems, either in a semi-supervised or fully automatic setting.

The remainder of this paper is organized as follows: in Section 2 we detail the proposed method, starting from single image segmentation, incorporation of spatial constraints and finally segmentation in a multi-view setting. In Section 3 we compare our method in a semi-supervised and fully automatic setting to existing methods [18,17]. In Section 4 we show the application of our method to accurate segmentation using a catadioptric camera system with planar mirrors, which greatly benefits from the information stemming from alternative views of the object. Additionally, we demonstrate how our method can improve the accuracy in multi-view background subtraction.

2. Spatially consistent multi-view segmentation

Starting with the binary segmentation of a single image, we show how to incorporate geometrical constraints among different views of an object. Our proposed segmentation method runs in interactive rates, so we also introduce the possibility for a user interactively refining the segmentation result.

2.1. Single view segmentation

Binary image segmentation is the problem of partitioning an image into disjoint foreground and background regions. Recently, very efficient globally optimal methods based on discrete graph cuts (e.g. [19]) and continuous variational methods (e.g. [20]) have been proposed. They allow to incorporate region terms and edge information to align the segmentation to the dominant edges in the image. In [21] we proposed a total variation regularized segmentation method, which we will use for single view segmentation.

In order to segment an image \( I \) we solve the following minimization problem:

\[
\inf_{u \in K^I} \left\{ \int_{\Omega} \sqrt{\nabla u^T D(\xi) \nabla u} \, d\xi + \lambda \int_{\Omega} uf \, d\xi \right\},
\]

(1)

where

\[
K^I = \left\{ u : \Omega \rightarrow [0,1] : \int_{\Omega} u \, d\xi \geq c \right\},
\]

(2)

and \( c : 0 \leq c \leq \int_{\Omega} \, d\xi \) is a global constraint which defines a minimum size of the segmentation. The image domain \( \Omega \) in our application is \( \mathbb{R}^2 \). The segmentation is given by \( u : \Omega \rightarrow [0,1] \) and \( f : \Omega \rightarrow \mathbb{R} \) is a potential function defining foreground and background respectively.

Fig. 1. Foreground and background probabilities for the input image (a). Probabilities obtained from a single view (b) and the probabilistic fusion of 13 views (c). Dark colors designate high foreground probability. Our segmentation result is overlaid in (a) and (c). The individual segmentations of a single view is shown in (b). Best viewed in color.
The first term of Eq. (1) is a weighted Total Variation regularizer which forces the segmentation to be smooth. The spatially varying weighting term has to be chosen such that the segmentation borders align with strong image gradients. In [21] we proposed two different choices for the local metric $D(x)$. In this work we use the anisotropic version, which encodes gradient magnitude and direction into the segmentation process. The anisotropic weighting not only proved to be useful for image segmentation, but has also been used in 3D reconstruction (e.g. [22]). The metric is defined as

$$D(x) = g(x)n \cdot n + n \cdot n - n \cdot n,$$

where $n = \sqrt{\sum n_i^2}$ is the tangent of $n$, $n_1 = nx \cdot n_0$ and $g(x)$ is an edge detection function, which is close to zero at strong edges. We do not know the size of the segmentation in advance, so we set $c = 0$ throughout the rest of the paper.

The second term in Eq. (1) is a data fidelity term, which allows us to guide the segmentation. The potential function $f$ should encode foreground and background information. Setting $f > 0$ results in $u$ tending toward positive values during the minimization process. $f > 0$ results in $u$ tending toward negative values. Since $u$ is bounded to $\{0, 1\}$, $f > 0$ encodes preferences for foreground ($u = 1$), whereas $f > 0$ pushes the segmentation toward the background ($u = 0$). $f = 0$ means that no information about the fore- or background is given in this area and the pure geodesic energy is minimized. We assume that the color distributions of the foreground and background are sufficiently different and apply $k$-component Gaussian Mixture Models (GMM) on the pixel values in RGB space, which have proven to be useful in segmentation before, in e.g. [19]. We define the probability of a pixel belonging to the color model defined by the GMM as

$$P_i(x) = \sum_k \omega_k N(l(x) | m_k, \Sigma_k),$$

where $N(l(x) | m_k, \Sigma_k)$ is a Gaussian distribution with mean $m_k$, covariance $\Sigma_k$ and mixture weight $\omega_k$. The potential function $f$ for image $i$ can thus be defined as negative log-likelihood

$$f_{\text{obj}}(x) = \log \left( \frac{P_{\text{obj}}(x)}{P_{\text{obj}}(x)} \right),$$

where $P_{\text{obj}}(x)$ and $P_{\text{obj}}(x)$ are probabilities for background and foreground, respectively.

The minimization problem is solved globally optimal via convex relaxation, by letting $u$ vary continuously in the interval $[0, 1]$, for $c = 0$, as shown in [21].

### 2.2. Spatial consistency

The Visual Hull, first formalized by Laurentini [8], is the largest shape which generates the same silhouettes in a number of views, as the ones actually observed. Consequently the projection of the visual hull into each input image perfectly overlaps the silhouettes used to create the visual hull. This is true for a perfect segmentation and camera calibration and is known as the silhouette consistency constraint [23]. Because every silhouette affects the visual hull and consequently its projection to all other views, segmentations in multiple views are not statistically independent.

Lee et al. [18] propose an automatic iterative segmentation method which uses the intermediate segmentation results from other viewpoints to estimate this silhouette consistency constraint. This measure has the nice property, that it is defined in image space, allowing us to formulate the geometric constraints between views without using an intermediate 3D grid. However, a drawback is that the segmentations have to be calculated iteratively to estimate the silhouette consistency.

Kolev et al. [17] avoid the iterative approach by fusing foreground and background probabilities instead of the binary silhouette information. In contrast to Lee et al. they carry out the segmentation in 3D space and obtain the silhouettes in the images by projecting the estimated surface. Here, the 3D space has to be discretized into a regular voxel grid. Each voxel is a assigned a foreground and background probability respectively:

$$P_{\text{obj}}(x) = \sum_{i=1}^{n} P_{\text{obj}}(x) \left( \prod_{j=1}^{n-1} (1-P_{\text{back}}(x)) \right),$$

where $\pi_i$ are the projections from voxel to pixel space. The probabilities $P_{\text{obj}}$ and $P_{\text{back}}$ are defined as $k$ component GMM as in Eq. (4). For a detailed derivation of Eq. (6) we refer the reader to [17]. In this work we assume a projective camera model which is given by:

$$P_i = K[R_i| t_i],$$

where $K$ defines the intrinsic parameters and $[R_i| t_i]$ the extrinsic parameters of the camera for viewpoint $i$. Additionally, lens distortion is removed from the images. Using this camera model, the projection from a voxel coordinate $X$ to the pixel coordinate system of viewpoint $i$ is defined by:

$$\pi_i(X) = P_i X.$$

The fused probability maps have shown to yield excellent reconstruction results. However, the segmentation is carried out in 3D which is quite expensive in terms of memory consumption and runtime. A simple reprojection of the reconstruction into image space usually does not yield pixel-accurate segmentations, especially for high-resolution images. Segmenting an object which projects to a size of $n \times n$ pixels would require approximately $n^2$ voxels to reach pixel accuracy. For $n > 500$ this quickly becomes infeasible. So, in the following section we will describe how we can use this voxel-based probabilistic model for segmentation in image space, independently for each input image.

#### 2.3. Multi-view segmentation

In order to transform the segmentation from 3D to 2D, the probability volume has to be projected into the input images. Similar to the single image color models, we define each element of the volume $V$ as

$$V(x) = \log \left( \frac{P_{\text{back}}(x)}{P_{\text{obj}}(x)} \right),$$

with $P_{\text{obj}}(x)$ and $P_{\text{obj}}(x)$ from Eq. (6).

Following the idea of silhouette constraints (the object has to project onto the silhouettes from which it was created), the potential function for the segmentation in each input image $i$ is defined as the maximum intensity projection (which equals the minimum if applied to a negative log-likelihood) of Eq. (9) into each input image:

$$f_{\text{obj}}(x) = \min \left\{ V \left( \pi_i^{-1}(x, \lambda) \right) \right\},$$

where $\pi_i^{-1}(x, \lambda)$ is the viewing ray for a pixel $x$ in image $i$:

$$\pi_i^{-1}(x, \lambda) = P_i X + \lambda t_i,$$

with $P_i^T$ being the pseudo-inverse of $P_i$ and $t_i$ the camera center. Fig. 1 shows a comparison of a potential function created from a single input image and from probabilistic fusion as described in Section 2.
Fig. 2 gives an overview on how the spatial constraints are created and how they are associated with each other.

Due to the projection we lose the information about the true surface position $\lambda$ along the ray $\pi^{-1}(x, \lambda)$. Therefore the result of Eq. (10) cannot be directly interpreted as the true log-likelihood for each pixel in the input image. However, the additional constraints for each image (Eq. (5)), which gave rise to the cost volume $V$ can now be compared to Eq. (10) in the pixel domain. We propose to fuse Eqs. (5) and (10) through:

$$f^{\text{fused}}_i(x) = \begin{cases} 
    f^{\text{si}}_i(x), & \text{sign}(f^{\text{sp}}_i(x)) = \text{sign}(f^{\text{si}}_i(x)) \\
    0, & \text{else}
\end{cases}$$

In that way, the spatial term $f^{\text{sp}}_i(x)$ is used as an indicator function, where the object is likely to be. In regions where spatial and the single image term agree upon the presence or absence of the object, the single image based probabilities are used. In contrast, in the other regions, the potential function $f^{\text{fused}}_i(x)$ is zero, which means that no information about the foreground and the background is available. Thus the segmentation in these regions will be solely influenced by the regularization term in Eq. (1), which forces the segmentation boundary to be aligned with the image gradients. Since the segmentation is still carried out in image space, it can be calculated globally optimal.

2.4. Incorporation of user input

A segmentation in the image domain allows a user to interactively interact with the method and get immediate feedback. We enable the user to define foreground and background regions which are then used to build the color models. The user can simply draw on the input image to define foreground and background regions. An intermediate segmentation result is presented within 100 ms, and can be refined through additional drawing. In addition, the user may define hard constraints which force the segmentation to be foreground or background, respectively. This kind of user input cannot be easily incorporated in the segmentation methods of [17,18], because these work in voxel space. Fig. 3 shows an example of user input and generated constraints for the segmentation.

2.5. Implementation notes

All steps of our method allow a parallel implementation. The fusion of the image color models in Eq. (6) and the maximum intensity projection in Eq. (10) can be calculated individually per voxel and pixel respectively. The minimization of the segmentation energy (Eq. (1)) can also be calculated on a per-pixel level. For the maximum intensity projection we used a standard ray casting with a step size set according to the voxel size and a nearest neighbor interpolation. We decided to implement the algorithms using NVidia’s CUDA framework.
On a system equipped with an Nvidia GTX480 graphics card, we are able to process roughly 180 million voxels per second. A single maximum intensity projection of a 2MP image takes around 25 ms. In a setup with 8 2MP camera views, the segmentation usually converges in less than one second.

3. Experiments

We evaluate our method using synthetic and real-world datasets. These datasets allow us to make statements about the accuracy of the obtained segmentation, because the ground truth segmentation is readily available. All datasets additionally include the fully calibrated cameras, which were used to create the images. We compare our method to other methods in two different settings: first, a completely automatic segmentation of an object, completely visible in all input images, and second, a semi-automatic segmentation, where a user may provide constraints for the foreground and background in one or more input images.

During all experiments we set \( \lambda = 1.0 \), \( g(x) = e^{-\beta|x|^p} \) with \( \beta = 15.0 \) and \( \alpha = 0.55 \). However, in our experiments we found out, that our method is not very sensitive to change these parameters. The voxel space is discretized with approximately 300 cubic voxels which easily fit in the memory of the used graphics card. The 3D bounding box is automatically determined as the common Field of View of all input cameras which see the object completely.

We use three different error measures which capture different aspects of the segmentation quality. The Dice Similarity Coefficient (DSC) measures the mutual overlap between two segmentations:

\[
\text{DSC}(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}, \tag{13}
\]

where \( X \) denotes the current binary segmentation mask and \( Y \) the ground-truth segmentation. DSC is a standard error measure for assessing the quality of a segmentation in the context of medical imaging. Similarly the hit rate measures the number of correctly classified foreground pixels:

\[
\text{HR}(X, Y) = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}. \tag{14}
\]

The false alarm rate measures the amount of pixels incorrectly classified as foreground:

\[
\text{FAR}(X, Y) = \frac{|X \setminus Y|}{|X| + |Y| - |X \cap Y|}. \tag{15}
\]

3.1. Semi-automatic segmentation

To assess the accuracy of our segmentation method we use multi-view datasets where a ground-truth segmentation is available.\(^1\) In this setting the user is allowed to define foreground and background regions from which the color models are derived. We compare our method to the 3D segmentation method of Kolev et al.\(^{[17]}\).

implementation of the authors was available, we reimplemented their algorithm. The parameters for this method are set according to the suggestion in the paper. To remove the influence of user interaction, both methods are given the same spatial constraints, generated from the same user interaction, as input.

In Table 1, the mean values of the error measures, defined in Eqs. (13–15) over all input images, are given for five different datasets. While the hit rate is comparable between the different approaches, the false alarm rate of the segmentations produced by our approach is significantly lower. As can be seen from this result, the mismatch of pixel and voxel resolution leads to an over segmentation in the approach of\(^{[17]}\), whereas our approach, working in pixel space, is able to find the exact boundary of the object. We performed additional experiments to show the influence of the spatial term. The columns in Table 1 labeled SV contain the segmentation results when using the potential function of Eq. (5) instead of Eq. (12), omitting the spatial information. The results substantiate the result of Kolev et al., who stated, that the segmentation using joint probabilities is favorable compared to independent segmentations in terms of accuracy. This holds also for our case, where the segmentation process itself is carried out independently, but on color models which encode the spatial dependencies of the images. To motivate the choice of our potential function (Eq. (12)), we ran an alternative experiment where we replaced Eq. (12) by Eq. (10) on the same datasets. The columns in Table 1 labeled MV2 show the result of this experiment. It turned out that our method achieved the same hit rate but also misclassified more pixels as foreground, therefore increasing the false alarm rate up to a factor of 6.

Fig. 4 shows the segmentation result for three of the input images in the bunny sequence. The first row depicts the input images, the second row shows the segmentation results when using independent color models, the third row shows the results with the additional spatial constraints. As can be clearly seen from this example, the spatial term helps to fill holes in the segmentation as well as suppresses the lights in the background, which feature the same color model as the foreground object.

Fig. 5 shows the segmentation result for some of the input images in the head sequence. The first row depicts the input images, and the second row shows the results using spatial constraints. As a comparison, the results using the method of\(^{[17]}\) are depicted in the third row. As it can be clearly seen, the difference in voxel and pixel resolution leads to a removal of small structures such as the stand on which the statue rests. Our method greatly benefits from the additional edge information extracted from the image gradients.

3.2. Comparison to multi-view reconstruction

For objects with distinct surface features it might be appealing to generate the segmentation by reconstructing the object using a state-of-the-art multi-view stereo algorithm like PMVS proposed by Furukawa and Ponce\(^{[25]}\) and backprojecting the reconstruction into the input images. Fig. 6(a) shows the result of such reconstruction with PMVS2 on the bunny sequence. As can be seen, not only the object, but also large parts of the background get correctly reconstructed. One

\(^1\) http://cvpr.in.tum.de/data/datasets/3dreconstruction.
of the input images and the projected reconstruction is depicted in Fig. 6(b and c). As can be seen from this example, in general the separation into object and background from an unordered point cloud is not trivial. Moreover, the reconstruction took over 25 min on a 8-core 2.6 GHz computer system, compared to less than ten seconds using our method on all 36 input images.

3.3. Segmentation of multiple objects

Our segmentation method is tailored toward a single object present in the scene. However, multiple objects can also be handled. In a qualitative experiment, we created a sequence of 8 images, showing a toy and a marker for camera pose estimation (Fig. 7). The user marked foreground and background regions in the first input image. Since both, the toy and the marker were set to foreground, both objects get segmented in all images. In one of the images a black lens cap is visible (Fig. 7(c)), which resembles the same color as the marker and therefore gets a high foreground probability (Fig. 7(d), black denotes foreground). Since the lens cap is only present in this image, it is not spatially consistent among the input sequence and is correctly identified as background.

3.4. Automatic segmentation

In some applications user interaction is not desired or feasible (e.g. when segmenting a long sequence of images). Under the assumption that the object is completely visible in all images, we are able to automatically segment the object:

Step 1. Determine the initial background region as a region which cannot be seen by all cameras.
Step 2. Initialize the pixel-wise foreground probability to be a uniform distribution.
Step 3. Using the background region, create a GMM to define the pixel-wise background probabilities in all input images.
Step 4. Generate the spatial constraints as in Eq. (12).
Step 5. Solve the segmentation by running the minimization of Eq. (1) for each image until convergence.
Step 6. Use the identified foreground and background regions to build GMMs and update the foreground and background probabilities respectively.
Step 7. Go to step 4. Repeat until convergence.

In a sense this strategy is similar to the ideas of Lee et al. [18]. The main difference to our approach lies in step 4. Lee et al. backproject the silhouettes obtained from the previous segmentation to update the background model in spatially coherent regions. However, in contrast to their approach we do not fuse the intermediate segmentation results, but the potential functions. Additionally, they do not create a GMM for the foreground in the iterative loop, but perform a single segmentation step with both, foreground and background GMMs as a post-processing step. We are able to show comparable segmentation accuracy on the Temple [26] and Bust datasets, using a single iteration of the above strategy. While the authors only used 10 and 6 images respectively, we tested our method on all available images in the datasets. The authors reported runtimes in the range of several minutes, whereas our method converges in less than 5 s for the whole Bust sequence. The detailed comparison is given in Table 2.

4. Applications

In this section we will show the versatility of our method by applying it to two different segmentation scenarios. First, a visual inspection task, which uses a special catadioptric lens to create multiple views of an object using a single camera image, and second, we use our method to improve the output of an arbitrary background subtraction method in a multi-view setting.

4.1. Accurate segmentation using a catadioptric camera system

The visual inspection of small objects in an industrial setting is essential for automated quality control. Usually, the objects have to be inspected from several viewpoints. A very elegant and at the
same time inexpensive approach is to use a catadioptric system. Reflective surfaces like mirrors are placed in the field of view of a camera to create additional views of a target object in a single image. Each mirror creates a virtual camera with a viewpoint behind its surface. Catadioptric lenses can be purchased as drop-in replacement of conventional lenses.\(^{3}\)

In our experiments we use a catadioptric camera system with 5 planar mirrors and a single 2MP camera, leading to a 6-view system in total. We calibrate it using the method of Gluckman and Nayar [27]. The virtual camera setup is depicted in Fig. 8. The camera observes the object from above, whereas the mirrors create virtual cameras around the object. The common viewing volume of the camera system is shown by the box in Fig. 8. The color models are initialized automatically from the center view. The first and second columns of Fig. 9 show the segmentation results with and without spatial information, respectively. Please note the improvement in the mirror views, where the color models alone do not allow a clean segmentation due to changes in illumination and shadows.

To test the influence of the voxel resolution on the resulting segmentation, we run our algorithm with voxel resolutions of 300³, 100³ and 50³, respectively. Each run is provided the same single image color models. Fig. 10 shows a comparison of the obtained segmentations. In Fig. 10(a–c) the output of the proposed segmentation method is shown. As one can see, the segmentation quality stays pixel-accurate, even for large mismatches in pixel and voxel resolution. The main reason for these high-accuracy segmentations is the fusion of 2D and 3D constraints and the subsequent segmentation in image space. Fig. 10(d–f) and shows a direct comparison of the discretization effect between a segmentation in 2D and 3D respectively for the same amount of voxels. The center view has been cropped to better show the effect. At a voxel resolution of 100³, the additional time for generating the spatial constraints is only 50 ms on the computer system described in Section 5.

### 4.2. Improving background subtraction in a multi-view setting

In the domains of human motion capture and surveillance, in addition to a set of calibrated camera views, accurate background information is required. Even in very constrained scenarios, simple background subtraction leads to non-robust segmentation results. Also more complex background models are usually applied on a per-image basis and do not incorporate the information of the complete camera configuration. In this experiment we show, how our segmentation method can be used as a post processing step to improve an arbitrary background subtraction method with negligible computational overhead. We use the publicly available dataset Dancer⁴ which consists of 8 video streams of a dancing woman, taken from hemispherically distributed

Table 2

Comparison of our segmentation method to that of Lee et al. [18] on several datasets with ground-truth segmentations. SV depicts the result from using the method of Gluckman and Nayar [27]. The virtual camera setup is depicted in Fig. 8. The camera observes the object from above, whereas the mirrors create virtual cameras around the object. The common viewing volume of the camera system is shown by the box in Fig. 8. The color models are initialized automatically from the center view. The first and second columns of Fig. 9 show the segmentation results with and without spatial information, respectively. Please note the improvement in the mirror views, where the color models alone do not allow a clean segmentation due to changes in illumination and shadows.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Images</th>
<th>DSC</th>
<th>HR</th>
<th>FAR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MV</td>
<td>SV</td>
<td>[18]</td>
</tr>
<tr>
<td>Temple</td>
<td>0.950</td>
<td>0.779</td>
<td>0.928</td>
<td>0.984</td>
</tr>
<tr>
<td>Bust</td>
<td>0.985</td>
<td>0.959</td>
<td>0.953</td>
<td>0.984</td>
</tr>
</tbody>
</table>

Fig. 6. (a) Multi-view stereo reconstruction of the bunny sequence. One of the input images (b) and the reprojected point cloud (c). As can be clearly seen from this example, the distinction into fore- and background points in the point cloud is a non-trivial task.

Fig. 7. Segmentation of multiple objects. Panels (a–c) show 3 of 8 input images of a self created image sequence, overlaid with the segmentation result. Panel (d) shows the single image foreground hypothesis of (c). The lens cap in the lower left corner is correctly identified as background because it only appears in one input image. See text for details.

---

⁴ http://charibdis.inrialpes.fr/.
cameras. In addition to the full camera calibration, background images are available. Since no ground-truth segmentations are available, we manually segment the first five frames using Adobe Photoshop.

In order to show the robustness of our segmentation method, we use segmentations created by a simple, single image background subtraction, as input. Having a foreground image \( I_f \) and a background image \( I_b \), the pixel-wise distance in an intensity normalized RGB space can be defined as

\[
d(x) = \left( \frac{1}{3} \sum_{c=1}^{3} \left( \frac{I_{f,c}(x) - \mu_f(x) - \alpha}{\nu_f(x) + \alpha} \right)^2 + \beta \frac{(I_{b,c}(x) - \mu_b(x))^2}{1 + \beta} \right)^{1/2},
\]

with

\[
n_f(x) = \left( \frac{1}{3} \sum_{c=1}^{3} I_{f,c}(x)^2 \right)^{1/2},
\]

\[
n_b(x) = \left( \frac{1}{3} \sum_{c=1}^{3} I_{b,c}(x)^2 \right)^{1/2}
\]

being intensity normalizers along the three channels of the color image. In our experiments we set \( \alpha = 0.1 \) and \( \beta = 1 \). Segmentation now simply consists of thresholding (Eq. (16)):

\[
u(x) = \begin{cases} 
1 & \text{if } d(x) > \theta \\ 
0 & \text{if } d(x) \leq \theta.
\end{cases}
\]

To calculate the spatial fusion term in Eq. (12), per-pixel foreground and background probabilities have to be known for each image. For an image \( t \) they can simply be defined as \( P_{obj}^t(x) = \mu_f(x) \) and \( P_{bck}^t(x) = 1 - \mu_f(x) \).

In Fig. 11 we varied the threshold in the interval \( \theta = [0.04, \ldots, 0.11] \) and compared the segmentation result for the first five frames of the sequence to the ground truth segmentation, using the error measures defined in Eqs. (13–15). The filled box plots depict the result obtained

**Fig. 8.** Setup of the virtual cameras in the used catadioptric camera system. The box depicts the common viewing volume of all cameras.

**Fig. 9.** Segmentation results in a catadioptric system. Segmentation results using single color models (first column) and fused color models (second column). The third column shows one mirror view in detail. The additional spatial information helped delineate the object border exactly.
from single image background subtraction, the hollow box plots from combining these segmentations via spatial fusion. Our method makes the background subtraction less dependent on the choice of the threshold and generally results in more accurate segmentations (higher DSC and HR as well as lower FAR). To visualize these numbers, Fig. 12 shows segmentations obtained from different thresholds in the first row for one of the input images. As can be seen from this simple example, it is not possible to select a single threshold which allows complete segmentation of the person, without segmenting some structures in the background. When the threshold is set too low, much of the background is wrongly detected as foreground (see Fig. 12(b)), resulting in a larger false alarm rate (FAR). By increasing the threshold, the FAR decreases, but at the same time foreground pixels are wrongly detected as background which results in a smaller hit rate (HR) (see Fig. 12(d)). However, using these segmentations as input to our method, results in the segmentations shown in Fig. 12(f–h). Our method preserves the fine details of the object and effectively reduces the amount of mislabeled pixels in the background. When using a voxel resolution of 50$^3$, the additional time for creation of the spatial constraints and subsequent segmentation is 50 ms, which allows the method to be used in a real-time application.

5. Discussion and conclusion

We have presented a multi-view segmentation method, which allows the fusion of single view information like color and gradient with geometric information stemming from multiple views of the same object. In contrast to existing methods, our algorithm delivers fast, pixel-accurate segmentations and thus can be used for applications where high accuracy is essential. A GPU implementation allows it to be used in automated as well as semi-supervised settings.

In our method, segmentation is carried out in image space, based on foreground probabilities which encode single image color information and the spatial dependencies among different views. This allows our method to generate accurate segmentations even for a big mismatch in pixel and voxel resolution, which makes the method very fast. It is important to note that our method is not limited to pure color information. In fact every model which is able to produce pixel-
wise foreground probabilities can be applied. For example, one could use the output of a classifier trained on dense image features (e.g. [28]).

In experiments, we were able to show that segmentation using joint probabilities is favorable compared to independent segmentations in terms of accuracy and robustness. Additionally, we showed that our method is generally faster and more accurate than segmentation in 3D space with subsequent reprojection. We successfully applied the method to multi-view segmentation in a catadioptric system and to real-time background modeling for motion capture.

A limitation of our method as well as the methods of [17,18] is, that camera views from all around the desired object should be available to produce reliable spatial consistency. Furthermore, spatial consistency cannot compensate for very bad single image constraints. In sequences, where the object cannot be described by the single view model in most images, the spatial fusion process will in general not help improve the segmentation result.

An open question for future work is whether temporal coherence can be integrated as an additional constraint, for example in background modeling, where the object appearance changes slowly over time. Since our method is independent of the applied color models, the investigation of more complex methods used to model color and texture of foreground and background might be beneficial to further improve the quality of the achieved segmentations.

Acknowledgments

This work was supported by the Austrian Research Promotion Agency (FFG) under the project SILHOUETTE (825843).

References


