N-Tuple Color Segmentation for Multi-View Silhouette Extraction

Abdelaziz Djelouah, Jean-Sébastien Franco, Edmond Boyer, François Leclerc, Patrick Pérez

N-tuple Color Segmentation for Multi-View Silhouette Extraction

Abdelaziz Djelouah[1,2*], Jean-Sébastien Franco[2], Edmond Boyer[2], François Le Clerc[1], and Patrick Pérez[1]

[1]Technicolor, Cesson Sevigne, France
http://www.technicolor.com

[2]LJK - INRIA Rhône-Alpes, France
http://morpheo.inrialpes.fr

Abstract. We present a new method to extract multiple segmentations of an object viewed by multiple cameras, given only the camera calibration. We introduce the n-tuple color model to express inter-view consistency when inferring in each view the foreground and background color models permitting the final segmentation. A color n-tuple is a set of pixel colors associated to the n projections of a 3D point. The first goal is set as finding the MAP estimate of background/foreground color models based on an arbitrary sample set of such n-tuples, such that samples are consistently classified, in a soft way, as "empty" if they project in the background of at least one view, or "occupied" if they project to foreground pixels in all views. An Expectation Maximization framework is then used to alternate between color models and soft classifications. In a final step, all views are segmented based on their attached color models. The approach is significantly simpler and faster than previous multi-view segmentation methods, while providing results of equivalent or better quality.

1 Introduction

Segmenting foreground objects in images is an important topic in computer vision with numerous applications in scene analysis and reconstruction. The problem has been extensively addressed in the monocular case, and in the multi-ocular case with controlled environments. Multi-view segmentation with general environments is however still a largely unsolved problem, despite the growing interest for multi-view systems and the potential of using multi-ocular cues in the segmentation process. In the monocular case, the segmentation is inherently ambiguous and requires *a priori* information, usually on the color model of the background or the foreground. Such information can come, for instance, from user inputs [1] or previous frames in a temporal sequence [2]. Intuitively, adding viewpoints should alleviate the need for prior knowledge, while still allowing improvement over monocular segmentation. This potential is still largely untapped,

* This work is sponsored by the Quaero Programme, funded by OSEO

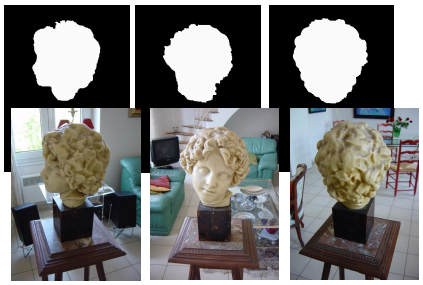


Fig. 1. Multi-view segmentation of foreground object with proposed method, using only minimal inter-view consistency assumptions and without resorting to 3D reconstruction.

in fact most multiple camera applications still rely on per-view segmentation, e.g. background subtraction, or on strong prior knowledge on the foreground, e.g. shape models.

Without prior information, the weakest inter-view assumption is the fact that consistent 2D foreground regions should define a single 3D region. Foreground is then defined as the spatially consistent region that has a color different from background and that appears entirely in all the views. A key difficulty in designing a multi-view segmentation algorithm is in how to enforce this inter-view consistency without sacrificing the simplicity of the approach. Indeed, we argue that the simplicity and efficiency is essential to the relevance and usability of the approach, as it is intended to be a pre-processing step to other algorithms further down in the analysis chain. While a small number of approaches exist specifically addressing the multi-view segmentation problem [3, 4], they usually involve quite complex pipelines. A number of methods [5–8] address multi-view silhouette extraction jointly with 3D reconstruction, relying on costly estimation of dense 3D representations, which we show to be unnecessary.

In this paper, we formalize the problem of inferring foreground/background color models, from which segmentation in each view is then readily obtained with classic tools, as a single, statistically sound model. By reasoning on the color n-tuple of an arbitrary 3D scene point, regardless of its actual position, we are able to formulate a simple generative model, which encodes the essential behavior of multi-view formation relevant to our segmentation goal. Contrary to existing approaches, we do not reason on dense 3D reconstructions, or voxel positions, but directly on an arbitrary sample set of such color n-tuples, inducing a drastic simplification with respect to previous models. The segmentation consequently translates into a simple EM algorithm, estimating per-view foreground/background color distributions and sample soft classifications consistent with our model.

2 Previous Work

Monocular Segmentation. Many approaches exist to monocular foreground / background segmentation. Low level background subtraction techniques reason at a per-pixel level, assuming a fixed or constant-color background has been observed with little corruption by foreground objects [9]. A number of such techniques also account for temporal changes of the background [10, 2]. The main advantage of these methods is computational efficiency, however the associated assumptions about background are often too strong to deal with general environments. More recent monocular techniques partially address this issue by formulating foreground extraction based on initial [11], or iteratively re-estimated [1] appearances of background and foreground, enforce spatial smoothness of the segmentations, using e.g. graph cuts. A drawback is in the semi-automatic nature of these algorithms, relying on manual input to distinguish foreground objects from the background.

Joint 2D and 3D Segmentation. A number of approaches treat multi-view silhouette extraction and 3D reconstruction simultaneously. We distinguish two sub-categories of methods here. The first category addresses primarily the 3D segmentation problem, treating silhouettes as noisy inputs from which to extract the best representation. A common feature we identify in this category is that they do not update and optimize per-view color models for foreground and background appearance. Solutions are found with well established convergence properties, e.g, using graph cuts [12], probabilistic frameworks [13], or convex minimization [5]. It is argued that consistent silhouettes are obtained as a by-product, generally by reprojecting the 3D reconstructions.

A second sub-category treats the joint 2D-3D problem include updates of color models for foreground and background [6–8]. This usually translates in a costly 3-stage pipeline, iteratively alternating between color models, image segmentations, and dense 3D visual hull representation. All resort to a form of conservative and costly binary decision of visual hull occupancy or 2D segmentation, e.g., using graph cuts in the volume [6], which we show to be unnecessary. Furthermore convergence properties of these pipelines are difficult to establish. In this paper, we demonstrate that multi-view silhouette segmentation can be obtained without a dense 3D reconstruction and therefore with the benefit of drastically reducing the complexity. Also, we show that our model is able to include color model updates while still keeping good convergence properties of EM, and using an automatic initialization. We compare our method with one of the most recent and successful Joint 2D/3D approaches [5].

Multi-view segmentation. The problem of multi-view foreground segmentation has only recently been addressed as a stand-alone topic, and few approaches exist. An initial work by Zeng et al. [3] identified the problem as finding a set of image segmentations consistent with a visual hull, and proposes an algorithm based on geometric elimination of superpixel regions, initialized to an over-segmentation of the silhouette. This deterministic solution proves of limited

robustness to inconsistently classified regions and still relies on an explicit 3D model.

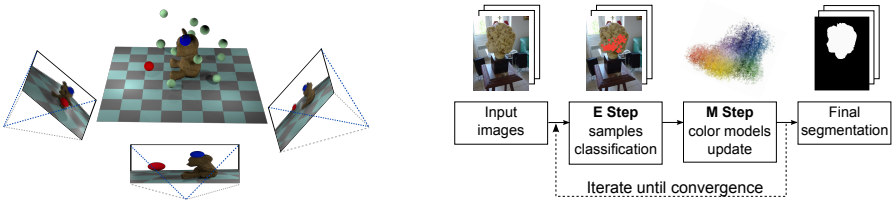
Some more recent approaches try to address the problem primarily in 2D using more robust, implicit visual hull representations, e.g., Lee *et al.* [4] give a probabilistic model of silhouette contributions to other images of pixels over their viewing lines, and alternatively update all views. A similar stance is taken by Sarim *et al.* [14], which infers segmentation as tri-maps by propagating information along epipolar bands. The proposed pipelines are still quite complex and fall just short of solving the 3D reconstruction itself. Convergence properties of these methods are hard to establish. [4] appears to be biased toward avoiding any under-segmentations to avoid irrevocably losing silhouette information, using conservative parameters / thresholds leading to slower convergence. We exhibit a new approach avoiding these defects using a 2D / 3D compromise, avoiding complete dense representations, while encoding the exact specificities of the multi-view segmentation problem.

3 Principle

Our purpose is to perform the foreground/background segmentation of a scene from n views obtained from a calibrated and synchronized camera setup. The regions that do not appear in the common field of view of all the cameras are considered background. Among the remaining parts of the scene, we define as foreground the regions of 3D space whose observed color distributions in the views differ from their background counterparts. The per-view color distributions of foreground and background are estimated following an approach similar to state-of-art single-view bilayer segmentation methods [1, 15]. We add an extra mechanism to foster the inter-view consistency of the segmentation. In contrast to most existing methods, we do not rely for this on an explicit dense geometry estimation of the 3D foreground regions. Instead, sparse 3D samples are used to accumulate and propagate foreground and background labels between views.

The key idea we propose to implement inter-image coherence is to consider color tuples $(I_1, \cdots, I_n)$ of the image set, taken at the respective image projections of a 3D sample point (see Fig. 2(a)). The intuition of the associated generative model is as follows. If a sample is from the foreground object, then all corresponding tuple colors should simultaneously be predicted from the foreground color model in their respective image. Conversely, if the sample is not from the foreground object, then there exists an image where the corresponding color of the sample should be predicted from the background color model in that image, all other colors of the sample being indifferent in that case. We thus assign each sample a classification variable k_s, with values in the state space $\mathcal{K} = \{\mathrm{f}, \mathrm{b}_1, \cdots, \mathrm{b}_\mathrm{n}\}$, where f is the foreground state, and b_i are the background states.

Importantly, once colors are assigned to 3D samples, all the reasoning is on the color tuples themselves. This is reassuringly analogous to many monocular segmentation algorithms, which classify pixels primarily according to their color.



(a) N-tuple sampling (b) Main Steps

Fig. 2. Method overview: (a) Samples are represented as spheres in 3D for viewing purposes. The blue sample is labeled as foreground, projecting in all foreground regions in the images. The red sample is labeled background because two cameras classify this sample as background. (b) Algorithm outline. The approach iterates between sample classification and color models update. A final foreground/background segmentation in the images is performed to transfer sparse sample classifications to dense pixels.

Furthermore, the 3D sample population is only supposed to well represent the variety of color co-occurences in 3D, not to densely sample the geometry in this space. This again differs from traditional approaches where unnecessary and complex geometric estimation interfere with color distribution estimation. The generative model and its iterative estimation are detailed in the following.

4 Modeling

4.1 Generative Model

Let $\mathcal{S}$ be the selected 3D sample set. The color n-tuple associated to the sample $s \in \mathcal{S}$ is $(I_s^1, \cdots, I_s^n)$ and $k_s \in \mathcal{K}$ is its classification label. We introduce a set of mixing coefficients π_k, representing the proportion of samples explained by each hypothesis in $\mathcal{K}$, to be learned by the model. We note Θ_i^c the parameters of the color distributions (background and foreground) associated with image i. The generative model is shown in Fig. 3, where each sample's color tuple is predicted according to its classification label k_s with priors π_k, and the global color models Θ_i^c. Interestingly, the model can be viewed as treating the multi-view n-tuple segmentation problem as a mixture of foreground-background models.

4.2 Color Models

A number of color models can be used for Θ_i^c's, such as Gaussian Mixture Models [10]. We denote by R_i the region of interest in image i that is assumed to contain all foreground parts. R_i can be computed automatically from the common field of views of the cameras, but could also be initialized more tightly on the foreground from user inputs. Drawing from recent monocular methods [15], we choose to represent distributions with histograms, to express the complementary nature of foreground and background distributions in the image. We use a simpler variant of the original idea, by noting that the number of occurrences in each

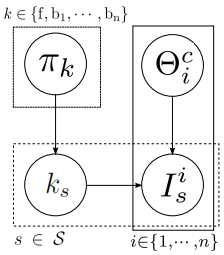


Fig. 3. Generative model : I_s^i, the color of the projection in the image i of the sample s, relates color models Θ_i^c according to its labeling k_s. π_k is the mixture coefficient.

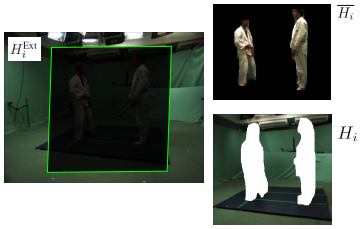


Fig. 4. The various color histograms, given an automatically selected region R_i that includes all foreground pixels in image i : H_i^{Ext} for known background region (R_i^c), H_i for background pixels inside R_i and $\overline{H}_i$ for foreground pixels.

bin of the background histograms (noted H_i) and foreground histograms (noted $\overline{H}_i$) of the region R_i sum to the number of bin occurrences of the whole region's histogram (noted H_i^{Int}). This enables to express bins of $\overline{H}_i$ as the difference between bins in H_i^{Int} and H_i (see Fig. 4). Both the foreground and background color models are thus fully parametrized by H_i, i.e., $\Theta^c = \{H_i\}_{i\in\{1,\cdots,n\}}$, since H_i^{Int} is given and fixed. Also, the complementary of region R_i in the image is initially identified as background, yielding a per-image histogram H_i^{Ext}. Such regions can be obtained automatically, typically by considering the projections in each image of the common visibility domain of all cameras, under the previously stated assumption that foreground objects are in this domain. To constrain H_i during initialization and convergence, we express that pixels of the outer region R_i^c, should also be well explained by H_i, leading to a prior term over H_i.

4.3 Joint Probability Distribution

Given the model, our goal is to find the parameters that maximize the *a posteriori* density given the observations. Noting $K = \{k_s\}_{s\in\mathcal{S}}$, $I = \{I_i^s\}_{s\in\mathcal{S},i\in\{1,\cdots,n\}}$, $\Theta^c = \{\Theta_i^c\}_{i\in\{1,\cdots,n\}}$ and $\pi = \{\pi_k\}_{k\in\mathcal{K}}$, the joint probability factorizes over samples s:

$$p(\Theta^c, I, \pi, K) = p(\Theta^c)p(\pi)\prod_{s\in S} p(k_s, I_s^1, ..., I_s^n|\Theta^c, \pi), \tag{1}$$

where $p(\pi)$ is uniform and will be ignored in the following steps. From Fig. 3, for a given sample s we have

$$p(k_s, I_s^1, ..., I_s^n|\Theta^c, \pi) = \left[\prod_i^n p(I_s^i|\Theta_i^c, k_s)\right] p(k_s|\pi). \tag{2}$$

If a sample is classified as foreground, then all colors from the corresponding tuple should be drawn from the foreground color model. But if a sample is

classified as background for the view i (label b_i) then the i-th color of the tuple should be predicted from the background color model in image i, and all other colors are indifferent, which we model as simply being drawn from the full image distribution:

$$p(I_s^i|\Theta_i^c, k_s) = \begin{cases} H_i(I_s^i) & \text{if} \quad k_s = \mathrm{b_i}, \\ \overline{H}_i(I_s^i) & \text{if} \quad k_s = \mathrm{f}, \\ H_i^{\mathrm{Int}}(I_s^i) & \text{otherwise}(k_s = b_j \mathrm{with} j \neq i). \end{cases} \tag{3}$$

This is really where the per view samples classification is performed. A sample satisfying background color model for a particular view i doesn't need to be checked against other color models in other views. It just needs to be likely under image histogram H^{Int}.

The term $p(k_s|\pi_{k_s})$ represents the mixture proportion prior

$$p(k_s|\pi_{k_s}) = \pi_{k_s}. \tag{4}$$

4.4 Prior From Known Background Pixels

We wish to enforce similarity between the distribution of background pixels and colors in the outer background region R_i^c. We model this by defining the prior over Θ^c as follows:

$$p(\Theta^c) = \prod_i \prod_{p \in R_i^c} H_i(I_p^i). \tag{5}$$

A set of given histogram parameters H_i is thus more likely if it explains known background pixels.

5 Estimation Algorithm

As the problem translates to a MAP estimation with latent variables, we use an Expectation Maximization algorithm. EM is an iterative process, which alternates between the posterior over classification variables given the current parameter estimate Φ^g (E-step), and estimating the new set of parameters Φ maximizing the expected log-posterior under the previously evaluated probabilities (M-step). In our case $\Phi = \{\Theta^c, \pi\}$. We build the E- and M-steps using the generically defined EM Q-functional, with established convergence properties [16]:

$$Q(\Phi, \Phi^g) = \sum_K \log(p(I, K, \Phi)) p(K|I, \Phi^g) \tag{6}$$

$$Q(\Phi, \Phi^g) = \sum_K \log(\prod_s p(k_s, I_s^1, ..., I_s^n|\Phi)) \prod_{s\prime} p(k_{s\prime}|I_{s\prime}^1, ..., I_{s\prime}^n, \Phi^g) + \sum_i \sum_{p \in R_i^c} \log(H_i(I_p^i)). \tag{7}$$

Simplifying this equation gives

$$\begin{aligned} Q(\Phi, \Phi^g) = \sum_s \sum_{k \in \mathcal{K}} \log\Big(p(k_s = k, I_s^1, ..., I_s^n|\Phi)\Big) p(k_s = k|I_s^1, ..., I_s^n, \Phi^g) \\ + \sum_i \sum_{p \in R_i^c} \log(H_i(I_p^i)). \end{aligned} \tag{8}$$

And the new set of parameters are $\Phi = \arg\max_\Phi Q(\Phi, \Phi^g)$.

5.1 Expectation Step

In the Expectation step, we compute for each sample s the probability of its classification hypothesis k_s:

$$\forall k \in K,\ p(k_s = k | I_s^1, ..., I_s^n, \Phi^g) = \frac{\pi_k^g \left[\prod_i^n p(I_s^i | \Theta_i^{g,c}, k_s = k) \right]}{\sum_z \pi_z^g \left[\prod_i^n p(I_s^i | \Theta_i^{g,c}, k_s = z) \right]}. \qquad \text{(noted } p_s^k\text{)} \tag{9}$$

5.2 Maximization Step

In this step, we find the new set of parameters Φ that maximizes the Q-function. We can write this function as the sum of independent terms:

$$Q(\Phi, \Phi^g) = \sum_{s,k} p_s^k \log \pi_k + \sum_i \left[\sum_{s,k} p_s^k \log(p(I_s^i | \Theta_i^c, k_s = k)) + \sum_{p \in R_i^c} \log(H_i(I_p^i)) \right]. \tag{10}$$

Each term can be maximized independently. For π_k:

$$\pi_k = \frac{1}{N} \sum_s p_s^k \qquad (\ N \text{ number of samples}) \tag{11}$$

Maximizing the view related terms is equivalent to maximizing

$$A_i(H_i) = \sum_s \left[p_s^{\mathrm{b_i}} \log(H_i(I_s^i)) + p_s^{\mathrm{f}} \log(\overline{H}_i(I_s^i)) \right] + \sum_{p \in R_i^c} \log(H_i(I_p^i)). \tag{12}$$

where we ignore the $\mathrm{b_j}$ labels ($j \neq i$) because they are related to the constant model H_i^{Int}. Let b be a particular bin in the color space. We note by H_b the number of occurrences in b for the histogram H. We can then write $A_i(H_i)$ as a sum of independent terms, each one related to a different bin of the color space:

$$A_i(H_i) = \sum_b \left[\sum_{\substack{s \in S \\ I_s^i \in b}} \left[p_s^{\mathrm{b_i}} \log(\tfrac{H_{i,b}}{|H_i|_{\mathrm{L1}}}) + p_s^{\mathrm{f}} \log(\tfrac{H_{i,b}^{\mathrm{Int}} - H_{i,b}}{|H_i^{\mathrm{Int}} - H_i|_{\mathrm{L1}}}) \right] + \sum_{\substack{I_p^i \in R_i^c \\ I_p^i \in b}} \log(\tfrac{H_{i,b}^{\mathrm{Ext}}}{|H_i^{\mathrm{Ext}}|_{\mathrm{L1}}}) \right], \tag{13}$$

It can be shown that optimizing this quantity is equivalent to updating bin values as follows:

$$H_{i,b} = \frac{\sum_{s \in S, I_s^i \in b} p_s^{\mathrm{b_i}} + H_{i,b}^{\mathrm{Ext}}}{\sum_{s \in S, I_s^i \in b} (p_s^{\mathrm{b_i}} + p_s^{\mathrm{f}}) + H_{i,b}^{\mathrm{Ext}}} H_{i,b}^{\mathrm{Int}}. \tag{14}$$

6 Final segmentation

The EM scheme described in the previous section, will converge to an estimate of the color models for each view and a classification probability table for each sample. The samples would only yield a sparse image segmentation if their classifications were crudely reprojected. This is why we use the obtained estimates to build a final dense 2D segmentation, combining results of sample classifications and color models. Note that this is only required after convergence in our approach, as opposed to being mandatory in the iteration with existing approaches. Segmentation amounts then to find for each pixel p of the ith view, the correct labeling l_p^i (foreground or background) according to the models (figure 6).

While various strategies could be used, we propose to finalize segmentation using a simple graph cut scheme similar to [11], minimizing a discrete energy:

$$E = \sum_p E_d(l_p^i|\Theta^s, \Theta_i^c, x_p, I_p^i) + \sum_{\{p,q\}\in N^i} \lambda E_s(I_p^i, I_q^i). \tag{15}$$

The data related term, E_d, at pixel p depends first, on how likely its color is under color models obtained for image i. It also depends on how its spatial position x_p relates to projections in the image of the set of softly classified 3D samples (Θ^s stands for the 3D samples' positions and associated probabilities $\{p_s^k\}_{s,k}$):

$$E_d(l_p^i|\Theta^s, \Theta_i^c, x_p, I_p^i) = -\log(p(x_p|\Theta^s, l_p^i)p(I_p^i|\Theta_i^c, l_p^i)), \tag{16}$$

- $p(x_p|\Theta^s, l_p^i)$ is proportional to a Gaussian around projections in the images of samples labeled foreground with a high probability. This allows to smoothly project inferred foreground information.
- $p(I_p^i|\Theta_i^c, l_p^i)$ is based on foreground or background histograms previously obtained:

$$p(I_p^i|\Theta_i^c, l_p^i) = \begin{cases} H_i(I_p^i) & \text{if} \quad l_p^i = \text{background}, \\ \overline{H}_i(I_p^i) & \text{if} \quad l_p^i = \text{foreground}. \end{cases} \tag{17}$$

E_s is the smoothness term over the set of neighbor pixels (N^i). It can be any energy that favors consistent labeling in homogeneous regions. In our implementation we use a simple inverse distance between neighbor pixels.

7 Experimental Results

In this section we present the experimental results we obtained using our approach in different situations. Experiments were done on synthetic and real calibrated multi-view datasets. We used 3D HSV color histograms, with 64 x 64 x 16 bins. In the initialization step, we are not making any assumption regarding the foreground/background proportion in image histograms. This means that background proportion in each bin of the image histogram is set to 0.5. To initialize the region of interest R_i, we use the common field of view but the method

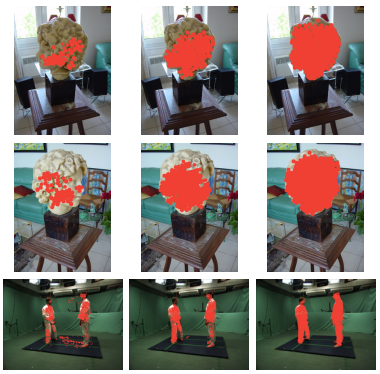


Fig. 5. Evolution of foreground samples on real datasets: red dots indicate the projection of the 3D samples from set S that have a high probability to belong to the foreground ($p_s^f > 0.8$)

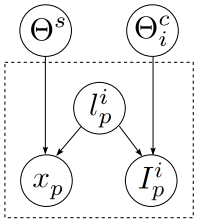


Fig. 6. Relation between variables in the final segmentation problem.

Fig. 7. Results with random sampling: original samples position, foreground samples and naive pixel classification after convergence.

is also entirely compatible with user inputs as it is shown in our experiments. Experiments were performed on a 2.0 Ghz dual core PC with 2GB RAM, with a sequential C++ implementation. Computation time is typically few seconds per iteration, and convergence was reached in less than 10 iterations for all the tests.

7.1 Space Sampling

Samples can be drawn from any relevant 3D point in space. In practice we draw samples from the common visibility domain of all cameras. This defines a bounding volume which is used to define regions R_i in each image i and find a first set of background pixels. For our initial experiments, we used a regular 3D sampling, and obtained very fast convergence for a small number of samples (50^3). More elaborate sampling schemes could be explored in future work, such as coarse-to-fine or adaptive samplings, which would further accelerate convergence. Fig. 5 illustrates the evolution of foreground sample classification probabilities during iterations on two datasets.

To emphasize that our approach doesn't need a regular sampling, we show results on TUM dataset[1] (Fig. 7) where we use a sparse random sampling (less than 10 000 samples). This sampling was enough to converge to a correct estimation of color models. In contrast, [5] on the same dataset and with the objective of joint estimation of silhouettes and geometry were using voxel grids with 200^3 or more voxels.

[1] http://cvpr.in.tum.de/data/datasets/3dreconstruction

7.2 Comparative Results

GrabCut We compare our segmentation results to a standard GrabCut segmentation to show the advantage of using multiview approach. The different results (Fig. 10 and Fig. 11) show typical GrabCut failure. In a monocular approach, it is hard to eliminate background colors that were not present outside the bounding box. In contrast, our approach benefits from the information of the other views and provides a correct segmentation.

Multiview based approaches We compared our approach with others multiview based methods like [4] and [5]. We used the publicly available kung-fu girl dataset[2] that was considered as challenging for [4]. The dataset consists of 25 calibrated views of a synthetic scene. We randomly selected 8 views for our experiments. Our algorithm converges quickly to the correct estimation of the foreground and produces near perfect segmentation after only 3 iterations (Fig. 8). The efficiency of our approach on this dataset, can be explained by the usage of 3D samples that allow to fuse all the information from the different views, but also by our compact parametrization of background and complementary foreground histograms.

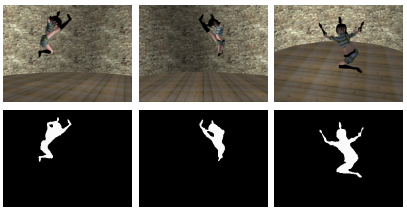


Fig. 8. Convergence after three iterations on the *kung fu girl* dataset.

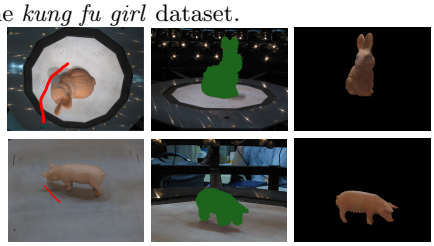


Fig. 9. Results on TUM dataset: red stroke indicates background region and green dots are foreground samples. Last column is the final segmentation.

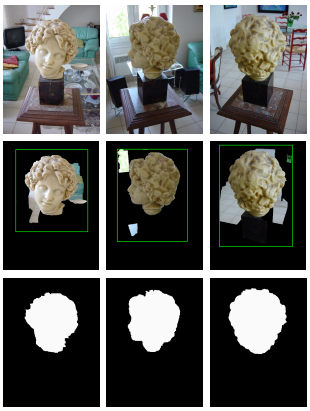


Fig. 10. Segmentation on the *bust* dataset: first row are input images, second row are results with grabCut and third row is the final segmentation with our method.

[2] http://www.mpi-inf.mpg.de/departments/irg3/kungfu/

We also tested our approach on the *Bust* multi view data set[3] and the *arts martiaux* datasets[4]. We selected the first dataset to give more comparison elements with Lee *et al.* [4] approach. The second dataset proves the ability of our approach to handle multiple foreground objects.

One of the advantages of the proposed estimation algorithm, is the fact that no hard decision is taken at any time. This means that samples once labeled as background with high probability, can be relabeled foreground during convergence if this is consistent in all the views, illustrating the increased stability with respect to existing approaches. An example of this can be observed on most of our experimental results (Fig. 11).

We also compare our approach with the approach proposed in [5]. In their approach, the main goal is the 3D reconstruction of the object. They use a bounding box around the object and user interaction in the form of two or three strokes in a given view, to have an initialization for foreground and background color models. Although our method proposes an automatic initialization we can also incorporate this type of prior. Typically with our method, one stroke in a single view is sufficient to propagate information to other views. For this dataset only two iterations were needed to convergence toward results identical to [5]. We obtain successful results using 40^3 samples (Fig. 9) and as low as 10 000 samples (Fig. 7).

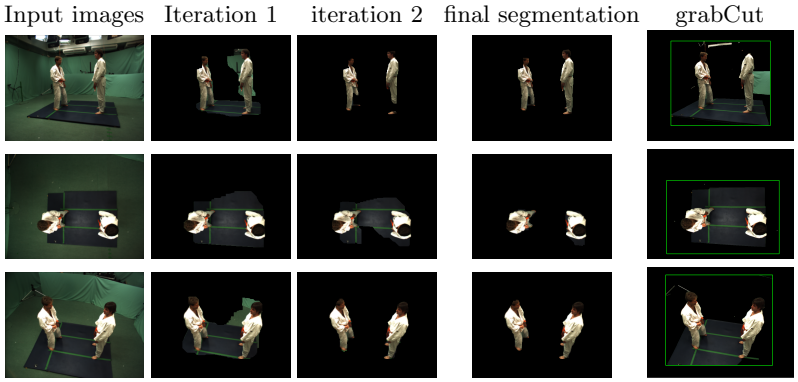


Fig. 11. Segmentation on *Art martiaux dataset.*

7.3 Results On Outdoor Datasets

We also tested our approach on challenging outdoor datasets[5] consisting in 4 views of a person in an urban environment. The method shows encouraging

[3] http://www.cs.ust.hk/ quan/WebPami/pami.html
[4] http://4drepository.inrialpes.fr/
[5] http://www.tnt.uni-hannover.de/papers/data/901/hb.tar.gz

results (Fig. 13), despite ambiguities in the color models of the foreground and background objects. Note again the model's ability to include the upper body after an initial background over-segmentation.

7.4 Convergence Rate

We illustrate the substantial convergence improvement by comparing to a state of the art method, Lee *et al.* [4] method (Fig. 12). Convergence is reached in a few seconds and fewer iterations, where [4] is mentioned to take several minutes. Although [4] does not indicate more specific times, the updates in their method (per-view per-pixel per epipolar line pixels in every other view) are more complex than ours, dominated by the E-step update of complexity $O(|S|n^2)$).

Regarding other comparable methods, [5] uses a fully optimized GPU implementation yielding per-iteration runtimes similar to our unoptimized CPU implementation. Note that our method could very easily be GPU optimized (embarrassingly parallel E- and M-steps) to achieve much higher speeds.

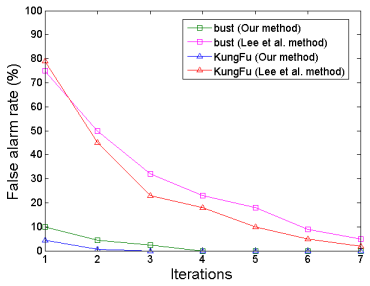


Fig. 12. Comparative convergence results with Lee et al. [4] approach on *Bust* and *kungfu girl* datasets.

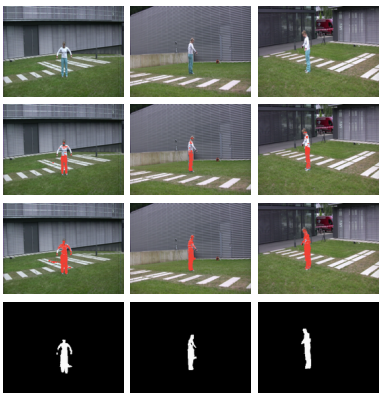


Fig. 13. Results on outdoor multi view dataset: convergence of samples labeling and image segmentation as final step.

8 Discussion and Future Directions

In this paper we have proposed a new simple and efficient approach to the multi-view segmentation problem. Experimental results suggest that our probabilistic formulation offers important convergence advantages over state of the art methods, and first experiments on outdoor sequences show promising results. Failure cases have been observed with configurations where the background and foreground color distributions are similar, a limitation common to methods characterizing regions using only color distributions. More discriminative models could

be used to improve the model in the future. Our method is successful using simple schemes for n-tuple sampling (regular grid, or random location in the region of common visibility), although thin objects (e.g. the subject's arms) can be missed if sampling is inadequate. More advanced sampling schemes could be proposed to yield more efficient and precise sampling.

Still, the method is largely successful, confirming previous findings and intuitions that useful segmentation can be obtained using only geometric, inter-view cues. This challenges the usual perception that only strong object priors can lead to perfect segmentations. While this may be true in the monocular domain, our work hints toward the possibility that multi-view cues, combined with a minimal number of additional weak cues, may prove sufficient to eliminate segmentation ambiguity for multi-camera setups.

References

1. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. In: ACM SIGGRAPH. (2004)
2. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. In: ICCV. (1999)
3. Zeng, G., Quan, L.: Silhouette extraction from multiple images of an unknown background. In: ACCV. (2004)
4. Lee, W., Woo, W., Boyer, E.: Silhouette Segmentation in Multiple Views. IEEE PAMI (2010)
5. Kolev, K., Brox, T., Cremers, D.: Fast joint estimation of silhouettes and dense 3d geometry from multiple images. IEEE PAMI (2011)
6. Campbell, N.D.F., Vogiatzis, G., Hernández, C., Cipolla, R.: Automatic 3d object segmentation in multiple views using volumetric graph-cuts. Image Vision Comput. (2010)
7. Feldmann, T., Dießelberg, L., Wörner, A.: Adaptive foreground/background segmentation using multiview silhouette fusion. In Denzler, J., Notni, G., Süße, H., eds.: DAGM-Symposium. Volume 5748 of LNCS., Springer (2009)
8. Gallego, J., Salvador, J., Casas, J., Pards, M.: Joint multi-view foreground segmentation and 3d reconstruction with tolerance loop. In: IEEE ICIP. (2011)
9. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. IEEE PAMI (1997)
10. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: CVPR. (1999)
11. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: ICCV. (2001)
12. Snow, D., Viola, P., Zabih, R.: Exact voxel occupancy with graph cuts. In: CVPR. (2000)
13. Franco, J.S., Boyer, E.: Fusion of Multi-View Silhouette Cues Using a Space Occupancy Grid. In: ICCV. (2005)
14. Sarim, M., Hilton, A., Guillemaut, J.Y., Kim, H., Takai, T.: Wide-baseline multiview video segmentation for 3d reconstruction. In: Proceedings of the 1st international workshop on 3D video processing. 3DVP'10 (2010)
15. Pham, V.Q., Takahashi, K., Naemura, T.: Foreground-background segmentation using iterated distribution matching. In: CVPR. (2011)
16. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)