Sparse Multi-View Consistency for Object Segmentation

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

Sparse Multi-View Consistency for Object Segmentation

Abdelaziz Djelouah[1,2], Jean-Sébastien Franco[2], Edmond Boyer[2], François Le Clerc[1], Patrick Pérez[1]
[1]Technicolor, Cesson Sévigné, France [2]LJK - INRIA Grenoble Rhône-Alpes, France

Abstract—Multiple view segmentation consists in segmenting objects simultaneously in several views. A key issue in that respect and compared to monocular settings is to ensure propagation of segmentation information between views while minimizing complexity and computational cost. In this work, we first investigate the idea that examining measurements at the projections of a sparse set of 3D points is sufficient to achieve this goal. The proposed algorithm softly assigns each of these 3D samples to the scene background if it projects on the background region in at least one view, or to the foreground if it projects on foreground region in all views. Second, we show how other modalities such as depth may be seamlessly integrated in the model and benefit the segmentation. The paper exposes a detailed set of experiments used to validate the algorithm, showing results comparable with the state of art, with reduced computational complexity. We also discuss the use of different modalities for specific situations, such as dealing with a low number of viewpoints or a scene with color ambiguities between foreground and background.

Index Terms—Segmentation, Scene analysis.

1 Introduction

Segmenting objects of interest is a first step for many applications in computer vision such as scene analysis, matting, compositing for post-production, image indexing and 3D reconstruction. Monocular segmentation requires priors about object shape or appearance, or user guidance [1]. However, in many situations, various viewpoints on the object are available and the exploitation of information across views enables to automate segmentation and to make it more robust as illustrated in recent works.

There exists two main families of methods to deal with this problem. First, cosegmentation approaches, which generally rely on shared appearance for the object of interest between views, high variability between background appearances across views, and separation of foreground and background appearance distributions. Second, silhouette-coherent extraction approaches, which rely on geometric consistency of the segmentations, often reconstructing some form of dense shape representation of the object of interest.

Both families have intrinsic limitations. The reliance of cosegmentation on appearance only [2] can be an obstacle when viewpoints are too further apart, causing drastic appearances discrepancy, whereas silhouette-coherent extraction approaches are often complex and computationally involved due to the updates of dense shape representations [3]. Both only rely on the color modality of video cameras and do not consider hybrid multiple camera systems using depth information where most of the research work has targeted scene reconstruction [4] and monocular segmentation [5].

In this paper we first present a full study of a new approach [6] that avoids altogether complete, dense shape representations, while encoding the specificities of the multi-view segmentation problem (see Fig.1 for a first example). Presented in Section 3, this approach is based on a complete probabilistic framework to account for geometric and appearance cues, allowing foreground/background segmentation without 3D object reconstruction or disparity map estimation. It allows us to cast the multi-view segmentation problem as Maximum A Posteriori (MAP) estimation over a sparse set of 3D samples. This estimation is classically achieved with an Expectation-Maximisation (EM) algorithm. The main steps of the approach are illustrated in Fig. 2. Using only multi-view color information, the approach proves simple and efficient. Its performance is thoroughly assessed in Section 4, with both qualitative and quantitative experiments and a focus on the influence of the number of views. The model is also flexible and can

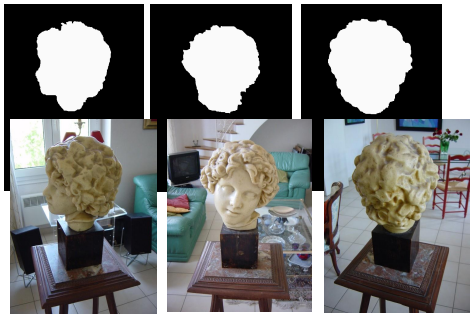


Fig. 1. Multi-view segmentation example of foreground object with the proposed automatic method, using sparse inter-view consistency constraints and without resorting to dense 3D reconstruction.

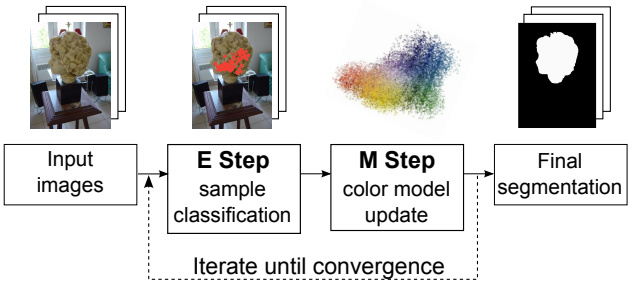


Fig. 2. Algorithm outline: The approach iterates between soft classification of sparse 3D samples and update of color models. A final foreground/background segmentation is performed in images to transfer sparse sample classification to dense pixel grid in each view.

be extended to embrace other types of measurements. We propose in particular to extend it to arbitrary set-ups with multiple depth and color cameras (Section 5), with no needs for pairing depth and color cameras, or rectifying them to a common viewpoint. We will show that in this case, only weaker priors are needed to isolate objects of interest automatically. In particular, we are able to extract segmentations even when there is no clear depth separation between foreground and background (Section 6).

2 Previous work

Prior work related to the figure-ground segmentation problem can be classified into four categories:

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 [7]. A number of such techniques also account for temporal changes of the background [8], [9]. 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 [10], or iteratively re-estimated [1] appearances of background and foreground and by enforcing 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 compensate for the inherent ambiguity of monocular segmentation.

Combining depth and color measurements in this category has also been investigated in several works, addressing the issues of foreground-background color ambiguity, light changes and occlusion. This was first experimented with stereo data [11], optimizing segmentation with color and depth region discontinuity terms. For complete automation, the methods usually need to introduce additional information defining the object of interest. A typical assumption is to identify frontmost depth regions as the object of interest, for simple extraction scenarios such as videoconferencing. This can be enforced using a depth threshold [12], or by considering two depth distributions for foreground and background [5]. This implies that the methods only work for good depth separation. In [13] user input is required and a segmentation manually defined in the first and the last frame is propagated using information from different modalities (*e.g.* range cameras, thermal camera).

Cosegmentation It was first coined in the seminal work of Rother *et al.* [2] as the simultaneous binary segmentation of image parts in an image pair and, by extension, to more images [14], [15], [16]. The key assumption of this family of methods is the observation of a common foreground region or object sharing appearance properties, versus a background with higher variability across images. The emphasis in these methods is often on the minimization of the distance between foreground histograms [2] or the maximization of their consistency [17]. Alternatively, discriminative clustering techniques can be used as in [15]. As noted by Vicente *et al.* [16], cosegmentation increasingly refers to a diverse set of assumptions and application scenarios, such as user-guided segmentation of large sets [14] or segmentation of object classes rather than a particular instance [18]. Given their specific hypotheses, cosegmentation approaches generally do not use geometric cues.

Segmentation from 3D reconstruction Unlike previously discussed approaches, the considered scenario is the same as in multi-view segmentation: multiple calibrated and synchronized views of the same scene. However, the primary objective of this category of methods is the 3D reconstruction and consistent silhouette segmentation are obtained as a byproduct, generally by reprojecting the estimated 3D reconstruction. These approaches rely either on the silhouette consistency of the visual hull [19], [20], [21] or on the photo-consistency of the photo-hull [22]. They assume some initial knowledge about foreground or background appearance, formulating the problem as a purely geometric extraction, using graph cuts [19], probabilistic frameworks [20], or convexification of the problem [21], with established algorithmic and convergence properties.

A number of works have also investigated using several depth cameras in conjunction with color cameras, targeting improvements in 3D surface reconstruction. Guan *et al.* [23] propose a probabilistic framework to volumetrically fuse depth and silhouette information. Kim *et al.* [4] propose a similar approach, this time adding a photoconsistency term, with the goal of improving multi-view stereo with depth information. Although in some cases it might be possible to extract

segmentations from the surface representation, clearly the aim of these works is 3D reconstruction: they do not specifically address how the different depth and color cues could be combined directly and efficiently for the purpose of reliably improving the segmentation in all views as proposed in this paper.

Multi-view segmentation The problem of multi-view foreground segmentation is increasingly addressed as a stand-alone topic and several methods have been proposed to segment an object seen in multiple views. Zeng *et al.* [24] first proposed a method based on classifying superpixel regions. Object silhouettes are identified as the union of a set of superpixel patches, each patch being iteratively examined and eliminated if inconsistent with the current object visual hull. While original, the proposed solution makes deterministic, hard decisions on patch labels and may diverge in case of any classification error. The work of [25] formulates the problem in a graph cut framework but requires short baselines to incorporate 2D shape coherence constraints between adjacent image pairs.

Very close to the previous category, some methods choose to build an explicit dense shape estimate, additionally re-estimating the parameters of color distributions of foreground and background regions, usually leading to complex and computationally intensive pipelines [3], [26], [27]. Reinbacher *et al.* [28] extend the work proposed in [21], addressing more specifically the multi-view segmentation problem with an iteration between a convex optimization and a color models update. All these methods share the common property of building an explicit, dense shape estimate, which we prove to be unnecessary in practice if the goal is only 2D silhouette segmentation.

Works more specifically addressing multi-view segmentation have proposed solutions to transfer information between views without an explicit 3D reconstruction. Still very close to 3D reconstruction, Lee *et al.* [29] focus on probabilistic occupancy along viewing lines. Their method iterates over each image, propagating occupancy information to other views. Kowdle *et al.* [30] present a method based on stereo and piece-wise planar reconstruction assuming short baseline viewpoints. In [31], Campbell *et al.* build a graph on a superpixel oversegmentation of the images. Superpixels from different views are then linked together using the epipolar geometry constraints (as in [32]). The method relies on the fixation condition to bootstrap the color model of the object and requires reliable stereo correspondences for good results.

In this paper we show that using a sparse set of 3D samples is sufficient to efficiently transfer the necessary information between the views. The proposed framework is also entirely compatible with hybrid camera systems where depth cameras and color cameras are simultaneously available for segmentation. To the best of our knowledge, this is not the case for any of the multi-view segmentation methods.

To avoid obscuring the discourse, we first present a detailed description of the method with only color cameras (§3) and its evaluation (§4). Then, the extension to deph data is explained (§5) and we will show in our experiments (§6), that using such information is relevant to resolve some color ambiguities that might arise in everyday sequences, making color-based segmentation impractical.

3 Proposed approach

The problem of multi-view segmentation can be intuitively defined as isolating objects of interest jointly seen in a set of views. According to [29], an object of interest should satisfy two constraints: be fully visible in all considered views, and have a general color appearance different from the background general appearance. This is the definition we use in this work.

In the proposed approach, bypassing the typical requirements of previous methods to compute a dense 3D representation of the object is made possible by focusing on a set of sparse 3D samples of the space commonly viewed by the n calibrated cameras and considering only the set of colors at the projections of each sample. The n colors present at pixel projections of a sample define a color n-tuple, which is the basic unit of information processed by the method. The spatial consistency of the foreground across views is expressed using these n-tuples. Since none of the 3D position related information is used (visibility, neighborhood, etc.), reasoning directly on these n-tuples allows a simpler and clearer framing of the problem. From now on, the terms *sample* and *n-tuple* will be indistinctly used to designate a 3D sample and its corresponding n-tuple, respectively. A generative model for sample labels is defined from the following intuition (Fig. 3). 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 images. This sample may not be visible in all the views but it will always project on the foreground region in the images. Conversely, if the sample is not from the foreground object, then there exists at least one image where the corresponding sample color should be predicted from the background color model in that image. This sample, that projects in a background region in one view, may project on background or foreground regions in the other views but will always be part of the images. So, this sample, predicted from the background distribution in one view, is predicted by the general image distribution in the other views. We note that this is in principle equivalent to deciding whether the 3D sample belongs to the visual hull of the object [29].

In the following, we describe the probabilistic model that relates in a consistent way the foreground and background models to the n-tuple colors. We use a maximum *a posteriori* approach to estimate both 3D sample states and appearance models.

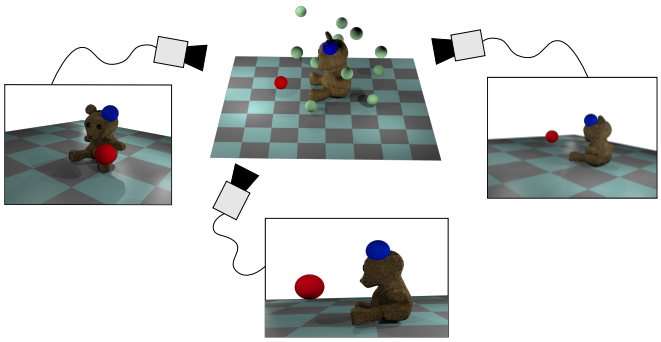


Fig. 3. Principle of multi-view object segmentation using sparse 3D samples: In this synthetic scene, the teddy bear satisfies our definition of *foreground object* and should be the result of our segmentation method. To identify the foreground region, samples (depicted by the spheres) are created in the common visibility volume. A sample is labeled as foreground (blue sphere), if it projects on foreground regions in all the views. In contrast, it is enough for a sample to project to background in one of the views to be labeled as background. This is the case here for the red sphere, classified as background as it projects to background in the middle and rightmost views, even though it projects to foreground in the leftmost view.

3.1 Probabilistic Model

Let $\mathcal{S}$ be the selected 3D sample set. The color n-tuple associated to the sample $s \in \mathcal{S}$ is $I^s_{1:n} = (I^s_1, \cdots, I^s_n)$. Following the intuition presented earlier, n-tuple colors should be predicted according to the sample state (foreground or background) and the appearance models. More precisely, for a foreground sample (labeled f) all its colors are predicted according to foreground shared appearance model Θ^F. At the opposite, a single view i is sufficient to label a sample as background (label b_i). In this case, the corresponding n-tuple color I^s_i is predicted according to the view specific color model Θ^B_i.

This reasoning is illustrated by the graphical model of Fig. 4, where each sample's color n-tuple is predicted according to its classification label k_s, and to the parameters Θ of the appearance models. The classification label k_s is in state space $\mathcal{K} = \{\mathrm{f}, \mathrm{b}_1, \cdots, \mathrm{b}_n\}$. The parameters π_k's are the mixing coefficients representing the proportion of samples explained by each hypothesis in k. They act as prior on the classification labels k_s. The proposed model can be viewed as a mixture of foreground-background models on the n-tuples where we try to estimate sample membership and foreground/background appearance models.

If we note by

- $I = \{I^s_{1:n}\}_{s\in\mathcal{S}}$ the set of image observations,
- $K = \{k_s\}_{s\in\mathcal{S}}$ the sample labels,
- Θ the appearance models,
- $\pi = \{\pi_k\}_{k,\in\mathcal{K}}$ the set of mixing coefficients,

our goal is to find the set of parameters $\Phi = (\Theta, \pi)$ that

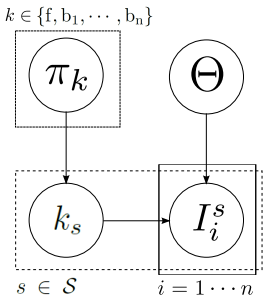


Fig. 4. Graphical model: I^s_i, the color of the projection in the image i of the sample s, relates color models Θ according to its labeling k_s. Parameter π_k is the mixture coefficient (label prior).

maximizes the *a posteriori* density given the observations:

$$\Phi = \underset{(\Theta,\pi)}{\arg\max} \quad \mathcal{L}(\Theta, \pi | I, K) p(\Theta, \pi), \tag{1}$$

where $\mathcal{L}(\Theta, \pi | I, K)$ denotes parameter likelihood. The MAP estimation of the parameters using only the observation is intractable. Therefore, as in other mixture fitting problems, unknown assignment labels K will be marginalized out (through EM) rather than explicitly estimated along with parameters.

Likelihood function. Given variable dependencies defined in our generative model, the likelihood function can be rewritten as follows:

$$\mathcal{L}(\Theta, \pi | I, K) = p(I, K | \Theta, \pi) = \prod_{s\in\mathcal{S}} p(k_s, I^s_{1:n} | \Theta, \pi), \tag{2}$$

where for a given sample s, assuming conditional independence of the observations in each view, we have:

$$p(k_s, I^s_{1:n} | \Theta, \pi) = p(k_s | \pi) \prod_{i=1}^{n} p(I^s_i | \Theta, k_s). \tag{3}$$

This is really where the per-view samples soft classification is performed. At this point, we formalize the definition of the foreground introduced earlier:

(A) *A foreground sample projects on foreground regions in all the views.* Using the shared foreground color model Θ^F, this translates to

$$\forall i \in [\![1, n]\!], \quad p(I^s_i | \Theta, k_s = \mathrm{f}) = p(I^s_i | \Theta^F). \tag{4}$$

(B) *One view is enough to label a sample as background.* For a sample s classified as background for the view i (label b_i), the i-th color of the n-tuple should be predicted from the background color model in image i. However, this sample can project in foreground or background regions in the other views. To model this in a simple consistent way, we use the distribution Θ^{Int}_j of the image region of interest R^{Int}_j. It avoids the estimation of sample visibility, while encoding both the projection in foreground and background regions. This translates to

$$\forall i \in [\![1, n]\!], \begin{cases} p(I^s_i | \Theta, k_s = \mathrm{b}_i) = p(I^s_i | \Theta^B_i), \\ p(I^s_j | \Theta, k_s = \mathrm{b}_i) = p(I^s_j | \Theta^{\mathrm{Int}}_j), \forall j \neq i. \end{cases} \tag{5}$$

Finally, the term $p(k_s|\pi)$ in Eq.3 represents the mixture proportion prior:

$$p(k_s|\pi) = \pi_{k_s}. \tag{6}$$

Prior from known background pixels. We denote by R_i^{Int} the region of interest in image i that is assumed to contain all foreground parts (see Fig. 5). Such a region R_i^{Int} can be automatically computed from the common field of views of the cameras [29]. With this assumption, the region of image i outside R_i^{Int}, hereafter noted R_i^{Ext}, must be the projection of the background, and can be used as a prior for the background appearance model.

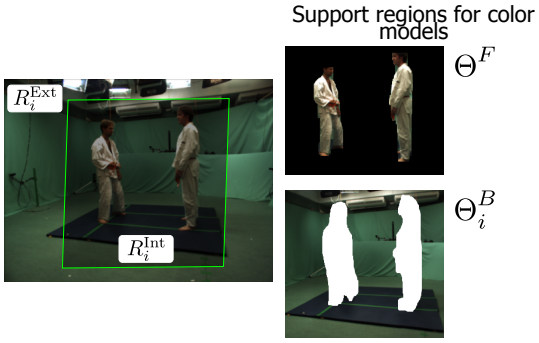


Fig. 5. Support regions for the color models using the assumption that foreground objects are seen in all images. R_i^{Int} is the projection of the common field of view that includes all foreground pixels in image i. Pixels in R_i^{Ext} are then known background pixels. Color model Θ_i^B is to be learned for background pixels inside R_i^{Int} and Θ^F is to be learned for foreground pixels (shared between the views).

One way to enforce similarity between the distribution of background pixels and colors in the outer background region R_i^{Ext}, with respect to our generative model, is to create 3D samples that project in this region and thus, have a background label.

We can define the following prior over Θ:

$$p(\Theta) = \prod_{i=1}^{n} \prod_{s \in \mathcal{S}_i} p(I_i^s|\Theta_i^B), \tag{7}$$

where $\mathcal{S}_i \subset \mathcal{S}$ is the set of such 3D samples relative to view i. A set of given appearance models $\{\Theta_i^B\}$ is thus more likely if it explains known background samples.

We can express the constraint in terms of pixels, instead of 3D samples:

$$p(\Theta) = \prod_{i=1}^{n} \prod_{p \in R_i^{\text{Ext}}} (p(I_i^p|\Theta_i^B))^{t_p}, \tag{8}$$

where I_i^p is the color of pixel p in image i and t_p is the number of 3D samples projecting onto this pixel.

Since we do not want to create samples outside the common field of view, we approximate the value of t_p with λ_i, the mean number of samples projecting on a single pixel in R_i^{Int}. The prior on the background color distribution is then the following:

$$p(\Theta) = \prod_{i=1}^{n} \prod_{p \in R_i^{\text{Ext}}} (p(I_i^p|\Theta_i^B))^{\lambda_i}. \tag{9}$$

3.2 Estimation Algorithm

The unknown parameters $\Phi = (\Theta, \pi)$ are obtained through MAP estimation:

$$\begin{aligned} \hat{\Phi} &= \arg\max_{(\Theta,\pi)} \quad \mathcal{L}(\Theta, \pi|I, K)p(\Theta, \pi) \\ &= \arg\max_{(\Theta,\pi)} \quad \prod_{s \in \mathcal{S}} \left[\prod_{i=1}^{n} p(I_i^s|\Theta, k_s) \right] \pi_{k_s} \\ &\qquad \cdot \prod_{i=1}^{n} \prod_{p \in R_i^{\text{Ext}}} (p(I_i^p|\Theta_i^B))^{\lambda_i}, \end{aligned} \tag{10}$$

where the classification labels k_s are treated as latent variables. We use an Expectation-Maximization algorithm that alternates between:

1) E-step: Computing the expectation of the posterior over the classification variables k_s, given the current parameter estimate $\Phi^g = (\Theta, \pi)^g$;
2) M-step: Estimating the new set of parameters $\Phi = (\Theta, \pi)$ maximizing the expected log-posterior.

We build the E- and M-steps using the generically defined EM Q-functional, with established convergence properties [33]:

$$Q(\Phi, \Phi^g) = \sum_{K \in \mathcal{K}^n} p(K|I, \Phi^g) \log \mathcal{L}(\Phi|I, K) + \log p(\Phi). \tag{11}$$

This EM Q-functional is the expected value of the log likelihood function, with respect to the conditional distribution of the hidden variables K, given the observations I and under the current estimate of the parameters Φ^g. Expanding each term of the sum, we get:

$$\begin{aligned} Q(\Phi, \Phi^g) = \quad & \sum_{K \in \mathcal{K}^n} \left[\log(\prod_{s \in \mathcal{S}} p(k_s, I_{1:n}^s|\Phi)) \prod_{s' \in \mathcal{S}} p(k_{s'}|I_{1:n}^{s'}, \Phi^g) \right] \\ & + \sum_{i=1}^{n} \lambda_i \Big(\sum_{p \in R_i^{\text{Ext}}} \log p(I_i^p|\Theta_i^B) \Big). \end{aligned} \tag{12}$$

Simplifying this equation gives [34]:

$$\begin{aligned} Q(\Phi, \Phi^g) = \quad & \sum_{s \in \mathcal{S}} \sum_{k \in \mathcal{K}} p(k_s = k|I_{1:n}^s, \Phi^g) \log p(k_s = k, I_{1:n}^s|\Phi) \\ & + \sum_{i=1}^{n} \lambda_i \Big(\sum_{p \in R_i^{\text{Ext}}} \log p(I_i^p|\Theta_i^B) \Big), \end{aligned} \tag{13}$$

and we can write this function as the sum of independent terms by summing over each label value:

$$\begin{aligned} Q(\Phi, \Phi^g) = \quad & \sum_{s,k} p_s^k \log \pi_k + \sum_{i,s} p_s^{\text{f}} \log p(I_i^s|\Theta^F, k_s = \text{f}) \\ & + \sum_i \Bigg[\sum_s p_s^{\text{b}_i} \log p(I_i^s|\Theta_i^B, k_s = \text{b}_i) \\ & + \lambda_i \sum_{p \in R_i^{\text{Ext}}} \log(p(I_i^p|\Theta_i^B) \Bigg] \\ & + \text{constant}. \end{aligned} \tag{14}$$

The constant term holds the contributions of labels b_j for $j \neq i$, which only depend on constant parameters Θ_i^{Int}. The new set of parameters is

$$\Phi^{g+1} = \arg\max_{\Phi} Q(\Phi, \Phi^g). \tag{15}$$

Expectation Step In the Expectation step, we compute for each sample $s \in \mathcal{S}$ the probability of its classification hypothesis k_s in the EM Q-function (Eq. 14):

$$\forall k \in \mathcal{K},$$
$$p_s^k := p(k_s = k | I_{1:n}^s, \Phi^g) = \frac{\pi_k^g \prod_{i=1}^{n} p(I_i^s | \Theta^g, k_s = k)}{\sum_{\ell \in \mathcal{K}} \pi_\ell^g \prod_{i=1}^{n} p(I_i^s | \Theta^g, k_s = \ell)}. \tag{16}$$

Maximization Step In this step, we find the new set of parameters Φ^{g+1} that maximizes the Q-function. Each term of the Q-function can be maximized independently.

For the mixture coefficients π_k, we use the Lagrange multiplier with the constraint $\sum_k \pi_k = 1$ (See supplementary material for details), and obtain the parameter update equation:

$$\pi_k = \frac{1}{N} \sum_{s \in \mathcal{S}} p_s^k \qquad (N\text{: number of samples}). \tag{17}$$

The appearance models have been defined in very general terms, and the equations derived so far are independent of the considered appearance models. We choose to represent color distributions Θ's using normalized histograms but any other color model can be used, including Gaussian Mixture Models [8].

M Step using color histograms. The background histogram for the view i is noted H_i. The shared foreground histogram is noted H^F. The region of interest R_i^{Int} in view i (Fig. 5) is also described by its histogram noted H_i^{Int}. All color models are thus fully parametrized by $\Theta = \{H^F, H_i, H_i^{\text{Int}} | i \in \{1, \cdots, n\}\}$. Sample labeling Equations 4 and 5 become

$$p(I_i^s | \Theta, k_s) = \begin{cases} H_i(I_i^s) & \text{if } k_s = \text{b}_i, \\ H^F(I_i^s) & \text{if } k_s = \text{f}, \\ H_i^{\text{Int}}(I_i^s) & \text{if } k_s = \text{b}_j \text{ and } i \neq j. \end{cases} \tag{18}$$

If we note $b \in [\![1, B]\!]$ a histogram bin and $H_{i,b}$ the value of b for histogram H_i and use the Lagrange multiplier with the constraint:

$$\sum_{b=1}^{B} H_{i,b} = 1, \tag{19}$$

solving $\Phi^{g+1} = \arg\max_\Phi Q(\Phi, \Phi^g)$ can be shown to come down to updating bin values as follows for the background (supplementary material for details):

$$H_{i,b} = \frac{\sum_{s \in \mathcal{S}: I_i^s \in b} \left(p_s^{\text{b}_i} + \lambda_i H_{i,b}^{\text{Ext}} \right)}{\sum_{b'=1}^{B} \sum_{s \in \mathcal{S}: I_i^s \in b'} \left(p_s^{\text{b}_i} + \lambda_i H_{i,b'}^{\text{Ext}} \right)}, \tag{20}$$

where $H_{i,b}^{\text{Ext}}$ is the number of pixels from R_i^{Ext} inside histogram bin b for view i. Likewise, the update equation for the foreground histogram reads

$$H_b^F = \frac{\sum_{i=1}^{n} \sum_{s \in \mathcal{S}: I_i^s \in b} p_s^{\text{f}}}{\sum_{b'=1}^{B} \sum_{i=1}^{n} \sum_{s \in \mathcal{S}: I_i^s \in b'} p_s^{\text{f}}}. \tag{21}$$

3.3 Final segmentation

The EM scheme described in the previous sections 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 of each image, 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 [28], [29]. Segmentation of view i then amounts to assigning to each pixel p of this image a binary label $l_i^p \in \{\text{f}, \text{b}\}$ (foreground or background) according to the current estimate of the color models Θ and to the set Ξ of the projection positions and label posterior probabilities of all the 3D samples (see Fig. 6).

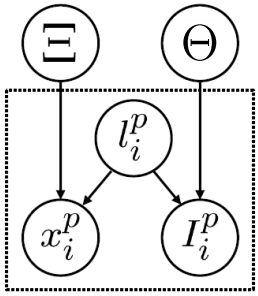


Fig. 6. Relation between variables in the final segmentation problem: l_i^p, x_i^p and I_i^p are respectively the binary label, the position and the color value of pixel p in image i. Variable Ξ stands for the 3D sample positions and associated posterior label probabilities. Variable Θ represents the foreground/background color model.

While various strategies could be used, we follow [29] and finalize segmentation using a simple graph cut scheme similar to [10], minimizing a discrete energy in view i:

$$E = \sum_p E_d(l_i^p | \Xi, \Theta, x_i^p, I_i^p) + \sum_{\{p,q\} \in N_i} \alpha E_s(l_i^p, l_i^q). \tag{22}$$

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 (Ξ stands for the 3D samples'

positions and associated probabilities $\{p_s^k\}_{s,k}$):

$$E_d(l_i^p|\Xi,\Theta,x_i^p,I_i^p) = -\log(p(x_i^p|\Xi,l_i^p)p(I_i^p|\Theta,l_i^p)), \quad (23)$$

where $p(x_i^p|\Xi,l_i^p)$ acts as prior from 3D samples:

- In the case $l_i^p = \mathrm{f}$, it is inversely proportional to the distance to the closest projection of a foreground sample. This allows smooth projection of inferred foreground information.
- In the case $l_i^p = \mathrm{b}$, this probability is constant (0.8 in our case to avoid disadvantaging foreground label which probability value is rarely equal to 1).

Second term $p(I_i^p|\Theta,l_i^p)$ is based on foreground or background histograms previously obtained:

$$p(I_i^p|\Theta,l_i^p) = \begin{cases} H_i(I_i^p) & \text{if} \quad l_i^p = \mathrm{b}, \\ H^F(I_i^p) & \text{if} \quad l_i^p = \mathrm{f}. \end{cases} \quad (24)$$

Second energy term E_s in (22) enforces the smoothness over the set N_i of neighbor pixels. It can be any energy that favors consistent labeling in homogeneous regions. In our implementation we use a simple inverse distance between neighbor pixels. A final remark is the possible inter-view inconsistencies in segmentation details due to the view-independence of segmentations in this final step.

4 Experiments with Color Images

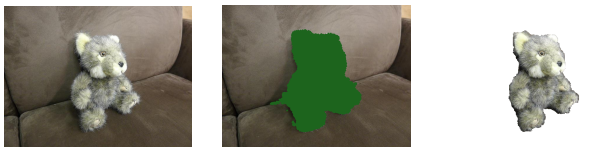
In this section we present segmentation results of the proposed method on various calibrated multi-view datasets summarized in Table 1. The objective is twofold: first, validate the n-tuple approach and, second, perform a comparative study with both monocular and multi-view segmentation approaches to show the improvements over the state of art. We also design some experiments to further investigate the sensitivity of the n-tuple model to the number of samples and to the number of viewpoints.

We use joint HSV color histograms, with $B = 32^3$ bins. Samples in $\mathcal{S}$ are drawn from the common visibility domain of all cameras. This defines a bounding volume which is used to define regions R_i^{Int} in each image i and to find a first set of background pixels, but the method is also entirely compatible with user inputs. For our initial experiments, we used a regular 3D sampling with about $N = 50^3$ samples. All the labels are set to the same probability for all the samples and we start the iterative process by a maximization step. We run our algorithm on a 2.5 GHz Intel Xeon PC with 12GB RAM, using a sequential C++ implementation.

4.1 Qualitative validation

For all the datasets, we show the foreground samples at convergence and segmentation results. The method performs almost perfectly on the *Bust*, *Couch* and *Bear* datasets (Figs. 1 , 7(a) and 7(b)). It can handle multiple foreground objects as in *Arts Martiaux* dataset. In Fig. 9, we show some of the intermediate results on the *Bust* dataset to illustrate algorithm convergence.

(a) *Couch* dataset: 10 views - Runtime $\approx 15s$



(b) *Bear* dataset: 10 views - Runtime $\approx 15s$

(c) *Car* dataset: 14 views - Runtime $\approx 30s$

(d) *Bike* dataset: 11 views - Runtime $\approx 20s$

Fig. 7. Results on datasets from [30]. We show (green dots) the projection of samples with high foreground probability ($p_s^{\mathrm{f}} > 0.8$) at convergence, and final segmentation.

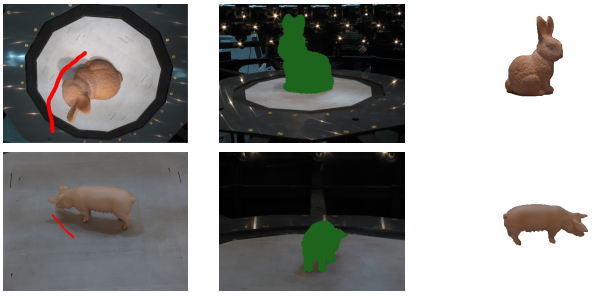


Fig. 8. Results on *Pig* and *Rabbit* [21]: The user indicates background region in one view (red stroke). Green dots indicate projection of samples with high foreground probability ($p_s^{\mathrm{f}} > 0.8$) at convergence. Last column is the final segmentation.

On *Car* and *Bike* (Fig. 7(c) and Fig. 7(d)), with significantly fewer points of view than state of art approaches, we achieve results of comparable quality. However in some views, the foreground/background color models are more ambiguous and this affects the segmentation results. This point is discussed in more detail in §4.3. Although our method proposes an automatic initialization, we can also incorporate user interaction. For example, on the *Rabbit* and *Pig* datasets the shadow on the ground is also seen by all the views and falls within our definition of the foreground. As in [21], user interaction is needed to resolve ambiguities. Typically with our method, one stroke in a single view is sufficient

Dataset	max. view number	User interact.	Comparison with GrabCut	Comparison with multi-view	Convergence study	Viewpoints influence	Influence of samples number
Arts Martiaux [6]	16		✓		✓	✓	✓
Bear [30]	15			✓	✓	✓	✓
Bike [30]	35			✓		✓	
Bust [35]	26			✓	✓		
Couch [30]	11		✓	✓	✓	✓	✓
Car [30]	44			✓		✓	
Pig [21]	27	✓		✓			
Rabbit [21]	27	✓		✓			

TABLE 1

The different calibrated multi-view datasets used. For each one, we indicate the original number of views and the tests we performed.

to propagate information to other views (Fig. 8).

To demonstrate the advantages of using a multi-view approach, we compare our approach with the *OpenCV* [36] implementation of GrabCut [1]. The GrabCut algorithm is initialized with a region of size equivalent or smaller than the region of interest used by our method. The results (Fig. 10) show that in a monocular approach, it is hard to eliminate background colors that are not present outside the bounding box. In contrast, our approach benefits from the information of the other views and provides a correct segmentation.

4.2 Quantitative evaluations

In this section we propose a quantitative evaluation of the proposed method, based on three performance metrics [29]: *mean error*, which gives a global measure of segmentation errors; *hit rate*, which indicates the proportion of well segmented foreground; and *false alarm rate*, which indicates the proportion of background segmented as foreground. Denoting W_b^a the subset of pixels from the set $a \in \{F, B\}$ (foreground or background) in the ground truth that are labeled as $b \in \{F, B\}$ by our segmentation, and $N(W_b^a)$ its cardinal, the performance metrics are defined as follows:

$$\textit{Mean Error} = \frac{N(W_F^B) + N(W_B^F)}{\textit{Number of pixels}}, \tag{25}$$

$$\textit{Hit Rate} = \frac{N(W_F^F)}{N(W_F^F) + N(W_B^F)}, \tag{26}$$

$$\textit{False Alarms} = \frac{N(W_F^B)}{N(W_F^B) + N(W_B^B)}. \tag{27}$$

We also define

$$\textit{Accuracy} = 1 - \textit{Mean Error}, \tag{28}$$

$$\textit{Missed Rate} = 1 - \textit{Hit Rate}. \tag{29}$$

Full quantitative evaluation of the method is proposed in Table 2. The mean and standard deviation are computed on segmentation results for all the views.

Number of views We also study the influence of the number of views used (Table 3 and Fig. 11) on four different datasets: *Bear, Couch, Bike* and *Car*. For a given number of views n, we randomly select n widespread views among those available in dataset and compute the segmentation. This test is performed 10 times for each number of views and the mean *Accuracy* is estimated. Using more views improve segmentation results. However, contrary to [30], using one third of dataset views is enough to produce good segmentation results (Table 3). We would like to emphasize the challenging nature of color ambiguities between foreground and background in *Car* and *Bike* datasets.

Dataset	Mean Error (%)	Hit Rate (%)	False Alarms (%)
Bust	0.2 ± 0.1	99.4 ± 0.01	0.7 ± 0.3
Arts Martiaux	0.5 ± 0.2	97.5 ± 0.3	2.7 ± 1.4
Couch	1.2 ± 0.8	97.0 ± 2.8	0.1 ± 0.1
Bear	2.7 ± 1.5	94.5 ± 3.0	7.0 ± 9.0
Car	2.8 ± 0.8	98.8 ± 0.8	16.7 ± 8.8
Bike	2.4 ± 1.1	96.7 ± 2.1	25.0 ± 13.3

TABLE 2

Full evaluation of the proposed approach on the different datasets.

	Our Method (a)	Our Method (b)	Kowdle [30]	Vicente [16]
Couch	7 98.7 ± 0.9	10 98.8 ± 0.8	10 99.6 ± 0.1	not available
Bear	5 97.3 ± 1.3	15 97.3 ± 1.5	15 98.8 ± 0.4	not available
Car	11 97.4 ± 0.8	44 97.2 ± 0.8	44 98.0 ± 0.7	44 91.4 ± 4.3
Bike	11 97.4 ± 1.5	35 97.6 ± 1.1	35 99.4 ± 0.4	35 88.9 ± 6.3

TABLE 3

Comparative results, using the proportion of correctly labeled pixels in the image (*Accuracy* in %). For each dataset, the number of views used is indicated.

Convergence of EM As a fully derived EM, our model converges to a local minimum. Our experiments show that silhouette-consistent objects (whose appearance is strongly different from the initial partial background) are strong easily reached minima (Fig. 12). The algorithm converges in 6 iterations for the considered datasets and even in 2 iterations for *Couch* and *Bear* datasets.

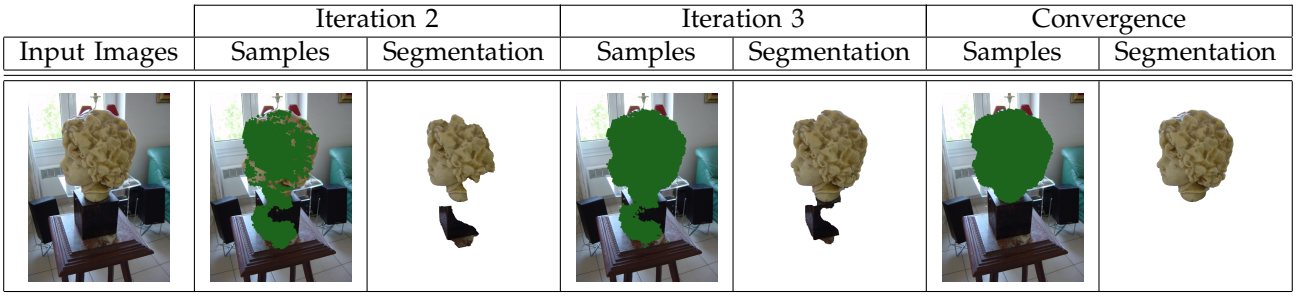


Fig. 9. Intermediate results of the algorithm on the *Bust* dataset ($n = 13$ views) with $N = 50000$ samples. Green dots indicate the projection of the 3D samples from set $\mathcal{S}$ with high foreground probability ($p_s^{\mathrm{f}} > 0.8$). Segmentation at each iteration is performed using the method described in §3.3. These intermediate segmentation results are used to study algorithm convergence (Fig. 12).

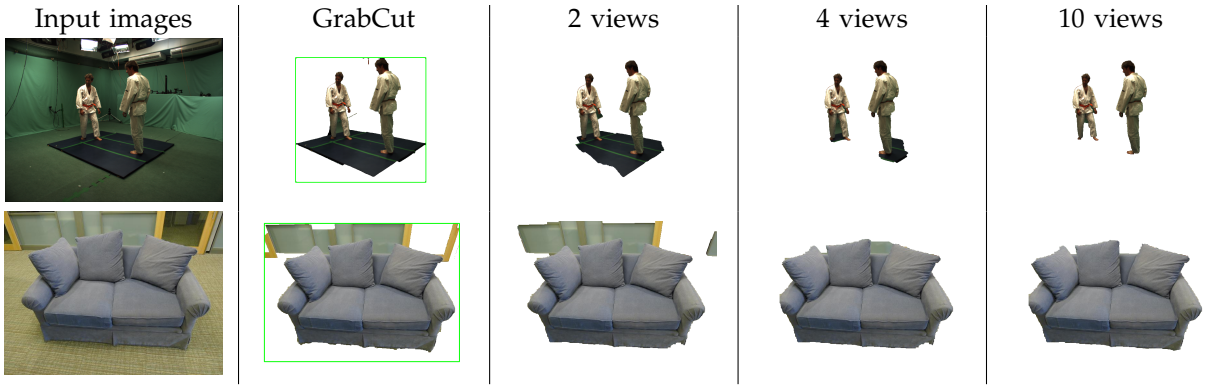


Fig. 10. Comparison with GrabCut monocular approach and influence of the number of views on our segmentation results on *Art martiaux* and *Couch* datasets.

Complexity and number of samples Each iteration complexity is linear in the number of samples and views, with running time of a few seconds (see Fig. 7 for details) where [30] indicates 2 minutes processing time on a single image due to the piece-wise planar depth map estimation and [29] indicates several minutes.

Grid based methods [21] and [28] show the same complexity properties but, as discussed in [28], the number of voxels is a key factor in the quality of segmentation: for an image resolution $M \times M$ a grid of $N = M^3$ voxels must be used to achieve pixel level precision. This becomes quickly unfeasible and, to circumvent the problem, they propose to perform segmentation at image level instead of directly taking grid projection. Still, the two methods rely on the same framework, using approximately 300^3 voxels and achieving reasonable runtime only through a GPU implementation. Thus, our method based on sparse sampling of the space, presents a key improvement over voxel based methods.

Fig. 13 shows *Missed rate* values for different numbers of samples on three datasets. In our model, using fewer samples implies using fewer colors. When colors are not used in the estimation, the prior from 3D sample projection in Eq. 22 becomes determinant in the segmentation. With a very sparse set of samples, the unknown colors are more likely to be labeled as background, inducing higher missed rate values. However, a random draw in the limited range of $20^3 \sim 50^3$ samples was enough to converge to a correct estimation of color models. This reflects directly on the processing time. For example, on the *Bust* dataset our non-optimized C++ implementation performs segmentation in 10s, while [28] report 5s with a highly optimized GPU implementation. We also note that our method is entirely compatible with a GPU implementation (highly parallel E- and M-steps). This would drastically reduce processing time and would be extremely gainful when extending the method to video sequences.

4.3 Discussion

The proposed model is built on the assumption that foreground and background objects have a different color appearance. This explains the results on *Bust* (Fig. 9) where the black objects in the background prevent the black base from being segmented as foreground. This assumption also implies that the foreground and background color models must be discriminative enough. This condition is hardly met when working on outdoor datasets. For example the *Car* and *Bike* datasets from [30] are really challenging due to the color ambiguity between foreground and background. The approach of [30] benefits from the short baseline between the views, which is used to estimate depths and more precisely a plane based reconstruction.

Another point to discuss is the final segmentation. As explained in section §3.3, various strategies can be used.

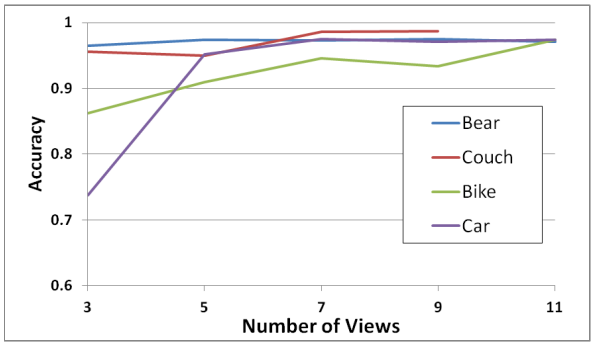


Fig. 11. Evolution of segmentation results (using *Accuracy*) according to the number of views used.

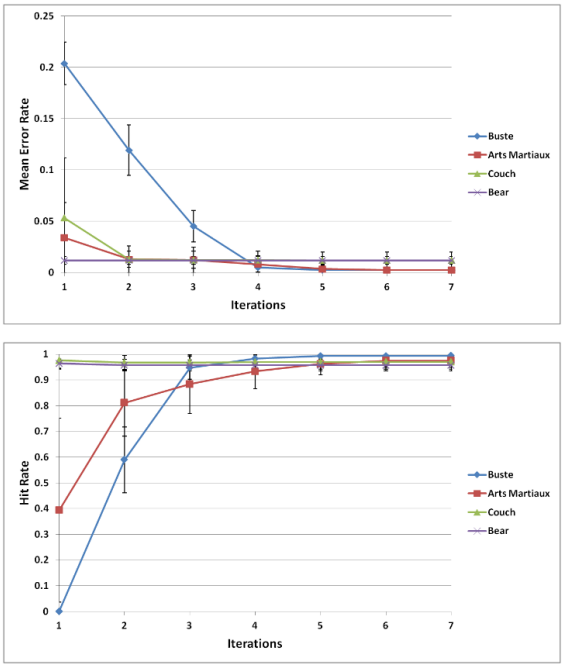


Fig. 12. Convergence study: Mean error rate and hit rate (with confidence intervals) on *Bust*, *Arts Martiaux*, *Couch* and *Bear*. For all the datasets, convergence is reached in 6 iterations. Only 2 iterations are needed for simpler scenarios like *Bear* and *Couch*.

In some cases, using a graph cut is not the best choice. Fig. 14 illustrates such a situation, where the samples are correctly labeled but the graph cut segments the thin parts as background. This leads to view inconsistent segmentations in the final step.

5 Combining Depth and Color Cues

The framework for multi-view foreground segmentation based on color observations presented and evaluated in the previous sections can be generalized to other cues. In this section, we extend the previously defined generative model to incorporate depth measurements from range cameras.

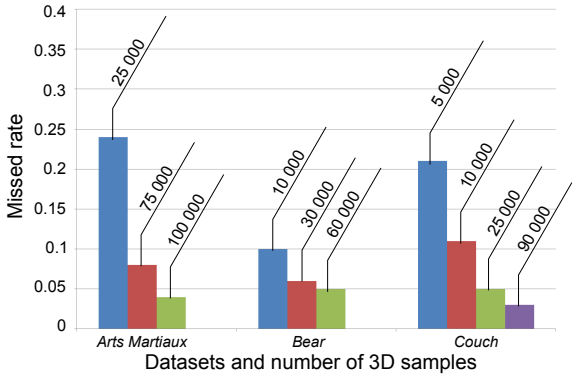


Fig. 13. Results with different numbers of 3D samples. Missed rate is used as the error measure (lower is better).

Fig. 14. Results on the *Chair* [30] dataset (using 4 views). Green points indicate projection of samples with high foreground probability. Last column is the final segmentation, with thin legs being lost while they were correctly labeled at the sparse samples level.

5.1 Principle

Range cameras provide measurements of the distance between the sensor and the scene objects. Even though these measurements are often noisy and sometimes locally inaccurate, they are informative of space occupancy in the scene and, in that respect, provide useful cues for the classification of samples in the presented framework.

Consider a sample s and a given range camera. Let d_s be the known ground truth distance of s to the center of this camera, z_s the depth provided by the range sensor in corresponding direction and $d_{\max}$ the maximum depth range. As identified by Guan *et al.* [23], the information that can be inferred on the occupancy of s by a foreground object depends on the relative values of d_s and z_s (See Fig. 15) :

- $z_s > d_s$: the sample is not occupied and should be classified as background.
- $z_s \simeq d_s$: this configuration corresponds to the highest probability that the sample is occupied by a foreground object.
- $z_s < d_s$: the sample lies behind an occluding object on its line of sight, nothing can be inferred about its occupancy.

These considerations should drive the choice of the space occupancy model from the depth observations and in some configurations, we may want to modulate the behavior of the algorithm in order to better adapt to the semantics of the scene. Consider for instance a scene with a can lying on a table. Though the table is a solid object inside this common visibility volume, because a significant part of the table lies outside of the common viewing volume, we may want the algorithm to single

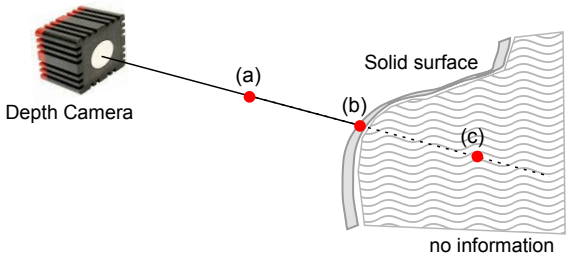


Fig. 15. Depth sampling situation on one projection line. Red dots indicate possible sampling position: (a) Depth measure is higher than sample distance to the camera ($z_s > d_s$). (b) Depth measure corresponds to sample position ($z_s \simeq d_s$). (c) Depth measure is lower than sample distance ($z_s < d_s$).

out only the can as foreground (See Fig. 19(a) for an example). We will show in the next sections that our modeling framework allows us to force either of these behaviors (select the table as foreground or background), by selecting an appropriate probabilistic model for space occupancy from the depth observations.

5.2 Depth-sensor enabled model

Let m be the number of depth maps and z_j^s the depth information associated with sample s in depth map j. We propose a new graphical model (Fig. 16), where the color tuple $I_{1:n}^s$ and the depth reading vector $z_{1:m}^s$ of each 3D sample $s \in \mathcal{S}$ are predicted according to its classification label k_s with priors π_k and the global color models Θ.

Posterior distribution Given the model, our goal is to find the parameters that maximize the *a posteriori* density given the observations. Noting $Z = \{z_{1:m}^s\}_{s\in\mathcal{S}}$, the likelihood function (Eq. 2) becomes

$$\begin{aligned}\mathcal{L}(\Theta,\pi|I,Z,K) &= p(I,Z,K|\Theta,\pi)\\ &= \prod_{s\in\mathcal{S}} p(k_s, I_{1:n}^s, z_{1:m}^s|\Theta,\pi),\end{aligned} \tag{30}$$

with:

$$\begin{aligned}&p(k_s, I_{1:n}^s, z_{1:m}^s|\Theta,\pi)\\ &\quad= p(k_s|\pi)\prod_{i=1}^{n} p(I_i^s|\Theta,k_s)\prod_{j=1}^{m} p(z_j^s|k_s),\end{aligned} \tag{31}$$

where color distributions $p(I_i^s|\Theta,k_s)$ are defined by (18) and depth distributions $p(z_j^s|k_s)$ will be defined below.

Estimation We follow the same EM scheme as in §4.2 to solve this MAP problem with latent variables. The main difference will appear at the expectation step (Eq. 16), where the new update of sample label posterior includes depth information:

$$\begin{aligned}&\forall k \in \mathcal{K},\quad p(k_s = |I_{1:n}^s,\ z_{1:m}^s, \Theta^g)\\ &\qquad= \frac{\pi_k^g \prod_{i=1}^{n} p(I_i^s|\Theta^g,k_s{=}k)\prod_{j=1}^{m} p(z_j^s|k_s{=}k)}{\sum_{\ell\in\mathcal{K}} \pi_\ell^g \left[\prod_{i=1}^{n} p(I_i^s|\Theta^g,k_s{=}\ell)\prod_{j=1}^{m} p(z_j^s|k_s{=}\ell)\right]}.\end{aligned} \tag{32}$$

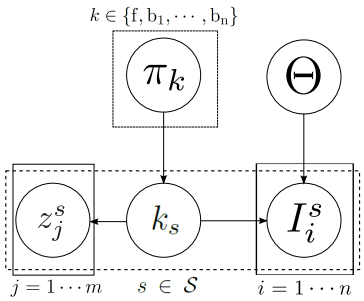


Fig. 16. Graphical model for color and depth: the generative model for color I_i^s and depth observations z_j^s are conditioned on the samples labels k_s.

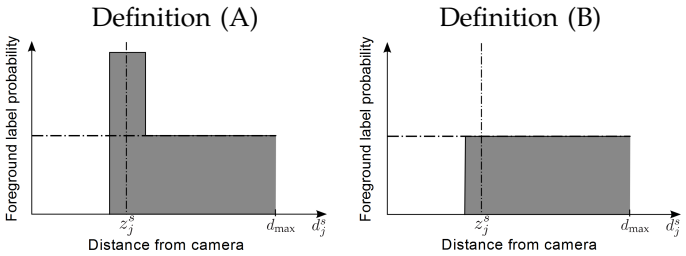


Fig. 17. Probability for a sample s to have a foreground label in the two situations A and B (See text for details). This probability depends on the sample distance from the camera d_j^s, the depth measure z_j^s and the maximum depth range $d_{\max}$.

The maximization step does not change (*i.e.*, equations 20 and 21). In the expectation step, terms related to the depth measurement act as priors on the samples labels. These terms can be computed once and for all in the initialization stage.

Modeling for depth Following the principles given in §5.1, our depth sensor model needs to classify as background all samples lying between the camera and the depth measurement along each line of sight and shouldn't give any information about samples behind front objects. This is expressed by giving, conditioned on f label, 0 probability to samples whose depth verifies $d_j^s < z_j^s - \varepsilon$, where ϵ is a conservative depth noise threshold. The sensor model behavior must also be defined when the position of the sample coincides with the depth measurement provided by the range camera. Different modeling possibilities exist and we empirically define distributions for two cases, following [23] and [4]:

(A) *Regions of space around the measured depth are more likely to contain an object than regions further away:*

$$p(z_j^s|k_s{=}\mathrm{f}) = \begin{cases} 1/d_{\max} & \text{if } z_j^s < d_j^s - \varepsilon,\\ (d_{\max} - d_j^s + \varepsilon)/(2\varepsilon d_{\max}) & \text{if } |d_j^s - z_j^s| \le \varepsilon,\\ 0 & \text{if } z_j^s > d_j^s + \varepsilon. \end{cases} \tag{33}$$

(B) *Regions further away than the measured depth are equally*

likely to contain an object:

$$p(z_j^s|k_s = \mathrm{f}) = \begin{cases} 1/(d_j^s + \varepsilon) & \text{if } z_j^s < d_j^s + \varepsilon, \\ 0 & \text{otherwise.} \end{cases} \quad (34)$$

For a background label b_i, the depth measurement is not informative and does not depend on these definitions:

$$p(z_j^s|k_s = \mathrm{b}_i) = 1/d_{\max}. \quad (35)$$

To give an intuition of what happens with these two different models, let's consider the simple situation of one depth camera, with all the priors and mixture coefficients set to uniform and a sample s, with its associated depth measure z_j^s. Fig. 17 shows the probability $p(k_s = \mathrm{f}|z_j^s)$ for this sample to have the foreground label according to its actual distance (d_j^s) from the camera in the two situations.

It turns out that first choice (Eq. 33) is detrimental to the segmentation performance when parts of the background are close to the foreground object, as illustrated on Fig. 18. In this case, samples on the table will have a high probability to be assigned foreground labels, owing to depth measurements. The colors at their projections in the views will be integrated in the foreground color model and will never be considered as background, despite the presence of similar colors in the known background region R_i^{Ext}. Consequently, in the sequel we will use the second model (Eq. 34).

Fig. 18. Regions around the measured depth are assumed to be foreground objects (Eq. 33). Green points are projection of samples with $p_s^{\mathrm{f}} > 0.8$.

6 Evaluation of Depth Contribution

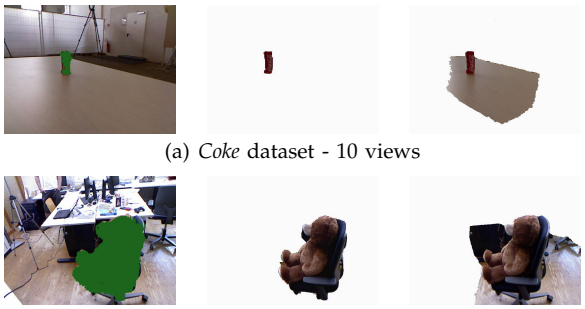
In this section, we show how the proposed method behaves in a multi-view context including depth cameras and how depth and color observations influence the results.

We choose to run tests on two different datasets. The first dataset was captured using a multi-view system consisting of $n = 2$ color cameras and $m = 2$ Swiss Ranger SR-4000 time-of-flight cameras (Fig. 20). The second dataset consists of three different Kinect video sequences[1]: *Coke*, *Plant* and *TeddyBear*. We select up to ten different frames from each sequence to constitute multi-view data-sets (see Figs. 19 and 21). We use the color model described in the previous section (except for the *Plant* dataset, see §6.2).

1. http://vision.in.tum.de/data/datasets/rgbd-dataset

6.1 Comparison with a monocular approach

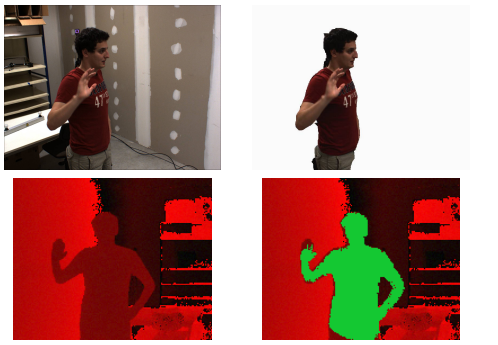
The results obtained by our approach on the considered datasets are shown on Figs. 19 and 20. A mean value of 5 iterations of the EM was needed to reach convergence. The method adapts well to the various configurations.



(a) *Coke* dataset - 10 views

(b) *TeddyBear* dataset - 8 views

Fig. 19. Results on Kinect datasets. Green dots indicate projections of samples labeled foreground with $p_s^{\mathrm{f}} > 0.8$. Second column is our segmentation and the third column shows results with our implementation of *TofCut* (see text for details).



(a) Results on *Boy1* dataset on color and range images

(b) Results on *Girl* and *Boy2*

Fig. 20. Results on multi-view datasets including $n = 2$ color cameras and $m = 2$ ToF cameras not aligned.

We compare our method with TofCut [5], a state-of-art monocular approach which describes foreground and background pixels using a weighted combination of color and depth models. The original algorithm was designed for datasets with a good discrimination in depth between foreground and background. In order to improve its robustness to overlapping depth distributions, we kept the same models but adopted an iterative

approach. Initially, pixels inside the region of interest R_i^{Int} are set to foreground, all other pixels are labeled background. Next, we alternate between pixel relabeling and model update, and iterate till convergence. Unlike the original TofCut algorithm, we chose to model foreground and background appearance and depth using histograms. This is of particular interest for depth, where the discrimination between foreground and background is not as strong as in [5]. Comparative results show that our approach consistently outperforms the modified TofCut method. Fig. 19(a) illustrates a typical failure case for monocular approaches such as TofCut, when the discrimination in depth between foreground and background is poor. Our method, in contrast, successfully handles the depth ambiguities, owing to the integration of information from all the views.

Our approach is also able to handle more complex acquisition situations where color and depth cameras are not aligned (Fig. 20), for which depth-based monocular approaches would not be applicable.

6.2 A more complex scenario - case study

As explained in §4.3, the difference in appearance between foreground and background is a key assumption of our approach. Not meeting this condition will lead to segmentation errors. With the *Plant* dataset, there is an ambiguity between foreground and background both in color and depth. If we define foreground to be any solid object inside the common visibility volume (using the model defined by equation 33), then the table will be segmented as foreground. If we add the assumption that the foreground must be visually different from the background, then only the green leaves are segmented as foreground and the blue pot is identified as background due to the blue objects in the background. In order to obtain a semantically meaningful segmentation, we use localized histograms to limit the propagation of background labels: the image is subdivided in rectangular regions, each region having its own histogram. Pixels participate in the histograms of the 4 closest regions. In this case, the combination of color and depth cues yields the expected segmentation (Fig. 21).

This dataset illustrates the complexity of the multi-view segmentation problem. It shows that color information is not enough to perform segmentation in complex scenarios and that depth information must be used with caution. This result also shows the adaptability of the n-tuple approach to various appearance models, which may be selected to better suit the desired segmentation semantics.

6.3 Quantitative results

We perform a quantitative comparison with ground truth to see how the combination of the two sources of information influences the results (Fig. 22). We compute false alarm and hit rates on segmentation results in three scenarios: color only, depth only and combining both.

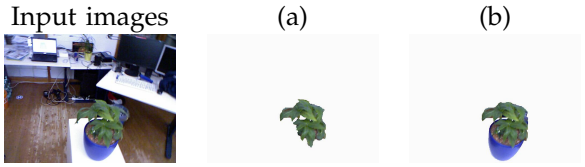


Fig. 21. Segmentation results on *Plant* dataset: (a) with a shared foreground color histogram; (b) with local histograms.

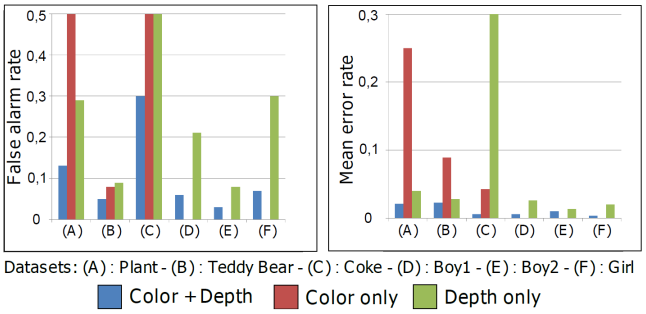


Fig. 22. Quantitative results: Comparison with ground truth in three configurations: depth only, color only and combining depth and color. Error rate gives a global measure for segmentation errors. False alarm rate indicates the proportion of background segmented as foreground. "Color only" is not given on last three sequences due to their insufficient number of image viewpoints ($n = 2$).

The color only method is sensitive to the resemblance between foreground and background. This explains the results on the Kinect datasets. It is also sensitive to the number of cameras. With only two color cameras, results on the second dataset are not representative and are therefore not shown on the figure. Results using only depth information are of better quality. The results on the *TeddyBear* dataset are very close to ground truth because all solid objects in the common visibility volume are parts of the foreground. However, using depth-only on the *Plant* dataset fails to correctly segment the table. Combining depth and color is very effective in these scenarios, where depth allows a quick identification and elimination of background regions and color allows a better accuracy of the foreground segmentation.

7 Discussion and Future Directions

We presented a new framework to solve the multi-view segmentation problem. This framework achieves results of equivalent quality to state of the art approaches, without being limited to short baseline scenarios and with substantially lower computational complexity. The method was successfully tested using hybrid set-ups made up of color cameras and depth sensors. It was shown to perform well, even in difficult configurations where depth discrimination between foreground and background is poor. Some failure cases still appear, when color and depth happen to be simultaneously undiscriminative, or when objects of interest do not fit the initial assumptions of the model (is the flower pot

part of the object of interest?). Still, the method is largely successful despite its use of weak inter-view cues, with no priors other than geometric. This challenges the usual perception that only strong object priors can lead to perfect segmentation. 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, making no assumption about the observed object, may prove sufficient to completely eliminate segmentation ambiguity.

REFERENCES

[1] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": interactive foreground extraction using iterated graph cuts," in *ACM SIGGRAPH*, 2004.

[2] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs," in *CVPR*, 2006.

[3] N. D. F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla, "Automatic 3d object segmentation in multiple views using volumetric graph-cuts," *Image Vision Comput.*, vol. 28, no. 1, pp. 4–25, 2010.

[4] Y. M. Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Micusik, and S. Thrun, "Multi-view image and tof sensor fusion for dense 3d reconstruction," in *3DIM*, 2009.

[5] L. Wang, C. Zhang, R. Yang, and C. Zhang, "Tofcut: Towards robust real-time foreground extraction using a time-of-flight camera," in *3DPVT*, 2010.

[6] A. Djelouah, J.-S. Franco, E. Boyer, F. Leclerc, and P. Pérez, "N-Tuple Color Segmentation for Multi-View Silhouette Extraction," in *ECCV*, 2012.

[7] C. Wen, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. PAMI*, vol. 19, no. 7, pp. 780–785, 1997.

[8] C. Stauffer and E. Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR*, 1999.

[9] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *ICCV*, 1999.

[10] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *ICCV*, 2001.

[11] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother, "Bi-layer segmentation of binocular stereo video," in *CVPR*, 2005.

[12] R. Crabb, C. Tracey, A. Puranik, and J. Davis, "Real-time foreground segmentation via range and color imaging," *CVPR Workshop*, 2008.

[13] J. van Baar, P. Beardsley, M. Pollefeys, and M. Gross, "Interactive video segmentation supported by multiple modalities, with an application to depth maps," in *3DTV-CON*, 2012.

[14] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "icoseg: Interactive co-segmentation with intelligent scribble guidance," in *CVPR*, 2010.

[15] A. Joulin, F. R. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *CVPR*, 2010.

[16] S. Vicente, C. Rother, and V. Kolmogorov, "Object cosegmentation," in *CVPR*, 2011.

[17] D. S. Hochbaum and V. Singh, "An efficient algorithm for co-segmentation," in *ICCV*, 2009.

[18] J. M. Winn and N. Jojic, "Locus: Learning object classes with unsupervised segmentation," in *ICCV*, 2005.

[19] D. Snow, P. Viola, and R. Zabih, "Exact voxel occupancy with graph cuts," in *CVPR*, 2000.

[20] J.-S. Franco and E. Boyer, "Fusion of Multi-View Silhouette Cues Using a Space Occupancy Grid," in *ICCV* , 2005.

[21] K. Kolev, T. Brox, and D. Cremers, "Fast joint estimation of silhouettes and dense 3d geometry from multiple images," *IEEE Trans. PAMI*, vol. 34, no. 3, pp. 493–505, 2011.

[22] K. Kutulakos and S. Seitz, "A theory of shape by space carving," in *ICCV*, 1999.

[23] L. Guan, J.-S. Franco, and M. Pollefeys, "3D Object Reconstruction with Heterogeneous Sensor Data," in *3DPVT*, 2008.

[24] G. Zeng and L. Quan, "Silhouette extraction from multiple images of an unknown background," in *ACCV*, 2004.

[25] M. Sormann, C. Zach, and K. Karner, "Graph cut based multiple view segmentation for 3d reconstruction," in *3DPVT*, 2006.

[26] T. Feldmann, L. Dießelberg, and A. Wörner, "Adaptive foreground/background segmentation using multiview silhouette fusion," in *DAGM-Symposium*, 2009.

[27] J. Gallego, J. Salvador, J. Casas, and M. Pards, "Joint multi-view foreground segmentation and 3d reconstruction with tolerance loop," in *ICIP*, 2011.

[28] C. Reinbacher, M. Rüther, and H. Bischof, "Fast variational multi-view segmentation through backprojection of spatial constraints," *Image and Vision Computing*, vol. 30, no. 11, pp. 797–807, 2012.

[29] W. Lee, W. Woo, and E. Boyer, "Silhouette Segmentation in Multiple Views," *IEEE Trans. PAMI*, vol. 33, no. 7, pp. 1429–1441, 2010.

[30] A. Kowdle, S. N. Sinha, and R. Szeliski, "Multiple view object cosegmentation using appearance and stereo cues." in *ECCV*, 2012.

[31] N. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla, "Automatic object segmentation from calibrated images," in *CVMP*, 2011.

[32] M. Sarim, A. Hilton, J.-Y. Guillemaut, H. Kim, and T. Takai, "Wide-baseline multi-view video segmentation for 3d reconstruction," in *3DVP*, 2010.

[33] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 2006.

[34] J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," ICSI, Tech. Rep., 1997.

[35] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Trans. PAMI*, vol. 27, no. 3, pp. 418–433, 2005.

[36] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

Abdelaziz Djelouah obtained a master degree in computer science from the university of Paris 7 in 2010. He is now a Ph.D. student at Technicolor and INRIA Grenoble. His research interests include computer visions areas such as multi-view segmentation, silhouette extraction and motion analysis.

Jean-Sébastien Franco is assistant professor of computer science at the Ensimag (School of Computer Science and Applied Mathematics, Grenoble Universities), and a researcher at the Inria Grenoble Rhône-Alpes and LJK lab, France, with the Morpheo team since 2010. He obtained his Ph.D. from the Institut National Polytechnique de Grenoble in 2005 with the Inria MOVI / Perception team. He started his professional career as a postdoctoral research assistant at the University of North Carolina's Computer Vision Group in 2006, and as assistant professor at the University of Bordeaux, with the IPARLA team, INRIA Bordeaux Sud-Ouest. His expertise is in the field of computer vision, with several internationally recognized contributions to dynamic 3D modeling from multiple views and 3D interaction.

Edmond Boyer obtained a Ph.D. in computer science from the Institut National Polytechnique de Lorraine in 1996. He started his professional career as a research assistant at the University of Cambridge (UK) in the Department of Engineering. Edmond Boyer joined the INRIA Grenoble Rhône-Alpes in 1998. From 1998 to 2010, he was associate professor of computer science at the university Joseph Fourier part of Grenoble universities. He is now a senior researcher at INRIA Grenoble Rhône-Alpes (France) where he leads the Morpheo research team on the capture and the analysis of moving shapes using visual cues. His fields of competence cover computer vision, computational geometry and virtual reality. His current research interests are on 3D dynamic modelling from images and videos, motion perception and analysis from videos, and immersive and interactive environments.

François Le Clerc graduated from the Ecole Supérieure dElectricité (Supélec) in 1988. From 1989 to 1994 he was with Thomson Underwater Acoustics research labs, where he worked on beam-forming techniques for active sonar imaging. In 1995 he joined Thomson Multimedia, now Technicolor, where he currently holds a senior researcher position. His research interests are in the fields of computer vision and machine learning, with applications to object segmentation and tracking, and facial image processing.

Patrick Pérez received the engineering degree from Ecole Centrale Paris in 1990, and the Ph.D. degree from University of Rennes in 1993. After one year as a postdoc in the Dpt of Applied Mathematics at Brown University (USA), he joined Inria (France) in 1994 as a full time researcher. From 2000 to 2004, he was with Microsoft Research (Cambridge, UK). He then returned to Inria as a senior researcher and took, in 2007, the direction of Vista research team of the Inria Rennes Center. In November 2009, he joined Technicolor where he leads exploratory research on computer vision and image analysis.