Shape from Silhouette in Space, Time and Light Domains

Maxim Mikhnevich and Denis Laurendeau
Computer Vision and Systems Laboratory, Laval University, Quebec, QC, Canada

Keywords: Shape From Silhouette, SFS, Multi-view Image Segmentation, Multi-lighting, Visual Hull, VH, Graph Cuts.

Abstract: This paper presents an image segmentation approach for obtaining a set of silhouettes along with the Visual Hull of an object observed from multiple viewpoints. The proposed approach can deal with mostly any type of appearance characteristics such as textured or textureless, shiny or lambertian surface reflectance, opaque or transparent objects. Compared to more classical methods for silhouette extraction from multiple views, for which certain assumptions are made on the object or scene, neither the background nor the object's appearance properties are modeled. The only assumption is the constancy of the unknown background at a given camera viewpoint while the object is under motion. The principal idea of the method is the estimation of the temporal evolution of each pixel over time which leads to the ability to estimate the background likelihood. Furthermore, the object is captured under different lighting conditions in order to cope with shadows. All the information from the space, time and lighting domains is merged based on a MRF framework and the constructed energy function is minimized via graph cuts.

1 INTRODUCTION

Shape from silhouette (SFS) is a classic computer vision technique for 3D object reconstruction. Exploring this technique in unknown environments when no prior information is available on the object's geometry or surface properties is still a difficult problem. The advantages of using the silhouette for reconstructing the shape of an object is that it requires neither constant object appearance nor the presence of textured regions.In the current work we exploit this property in order to reconstruct the Visual Hull (VH) of a wide set of objects.

More precisely, the aim of our work is to extract the silhouette of an object from a set of views without prior knowledge of the scene content or the object properties such as appearance and geometry and use these silhouettes to build a VH. This task faces several challenges. Firstly, the object interaction with light includes many effects such as shadows, self-shadows, color bleeding, light inter-reflection, transparency and subsurface scattering. These phenomena have an impact on the appearance of the object in the image and make the separation of foreground from background a complex task. Secondly, the camera can be positioned at any viewpoint on the hemisphere above the object, which leads to the impossibility to model the background at the pixel level (as done previously in static (Snow et al., 2000) or active (Matusik et al., 2002) cases) before positioning the camera even if the viewpoints are calibrated. Finally, the scene being captured under unknown lighting conditions adds extra complexity to the silhouette extraction problem. To cope with these phenomena we propose a fundamental approach where the object moves in an unknown but static environment while the camera remains fixed for a given viewpoint. The only assumption that is made about the scene is one on the background being static while the object moves. In comparison to other approaches which consider that the scene's background is known beforehand or which assume object photometric consistency, the proposed approach does not make any assumption about the object and therefore allows the handling of a wide variety of objects with surface reflectance properties ranging from textureless to completely transparent.

The experiment is performed as follows: the object is placed on a turntable which is then rotated in order to capture the object from different viewpoints. The images captured with this procedure are processed in a time sequential manner. Assuming a constant background, the time sequence is analyzed and the *background likelihood* is estimated. Then, the *object likelihood* is iteratively updated in order to estimate object boundaries precisely. Finally, several time frames are processed simultaneously to enforce boundary consistency between frames. All the computations are based on a Markov Random Field

Mikhnevich M. and Laurendeau D..
Shape from Silhouette in Space, Time and Light Domains.
DOI: 10.5220/0004722403680377
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 368-377
ISBN: 978-989-758-009-3

(MRF) framework and the optimization is performed through graph cuts. The silhouettes obtained for all viewpoints are used to build the VH of the object.

The paper is organized as follows: in section 2 an overview of the related work is given. Section 3 introduces research hypotheses and the notation used in the paper. In sections 4-6, the details of the estimation of background and object likelihoods as well as the segmentation framework are presented. Section 7 presents the experiments and discusses the results. The last section provides some conclusions on the proposed approach and identifies directions for future work.

2 RELATED WORK

SFS was first introduced by Baumgart (Baumgart, 1974), this concept suggests to fuse silhouettes of an object in 3D to obtain the VH. Since the object's silhouette is the key element for VH construction, the following review concentrates on silhouette extraction approaches.

The obvious and easy way to implement techniques for silhouette extraction is chroma keying (Smith and Blinn, 1996). This approach is based on the knowledge of the scene background. An object is imaged against a uniform or known background, then the silhouette is extracted by thresholding the background color or by background subtraction. Due to implementation simplicity, this technique was used in many SFS works (Matusik et al., 2002; Jagers et al., 2008). Even though this method provides fairly good results, there are some drawbacks. Firstly, it implies preliminary scene background manipulations for each camera viewpoint, which limits possible camera positions on a hemisphere since the background has to be visible from all viewpoints. Secondly, the case when part of the object has the same color as the background may lead to incorrect segmentation.

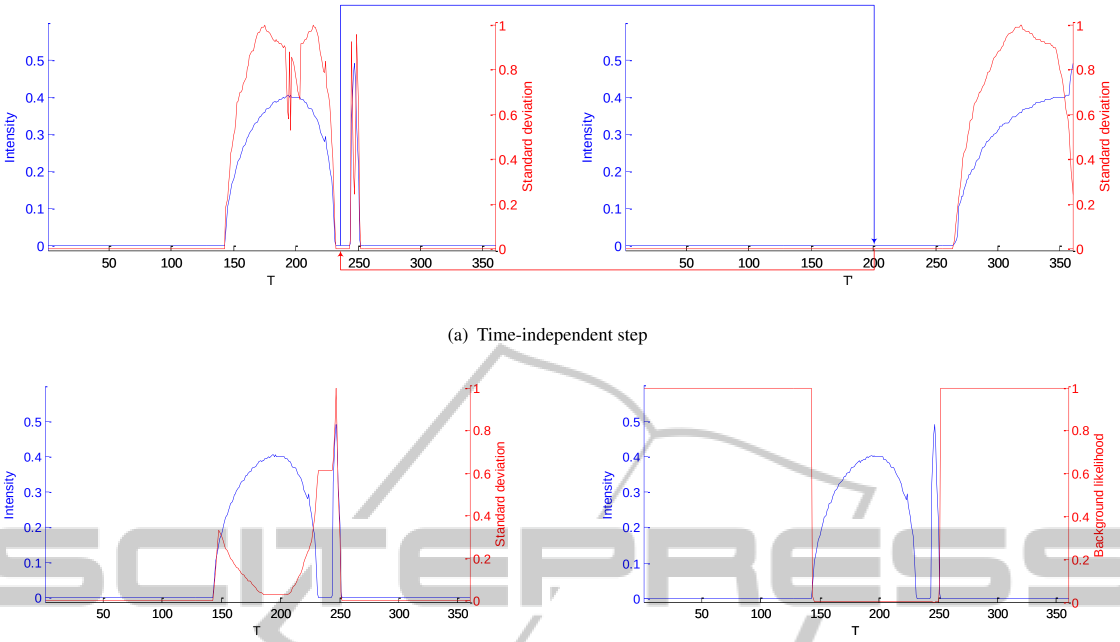
Chroma keying was extended in other works where instead of a static background, an active background system was used (Zongker et al., 1999; Matusik et al., 2002). As an active background, a controlled display was installed around an object. A scene was captured with and without an object with different background patterns for a fixed viewpoint. Even though such an approach allows the extraction of the alpha matte of the silhouette of an object made from material with complex optical properties such as glass, the hardware requirement seriously complicates the acquisition process and limits the method's application area. The major drawback is the inability to move the camera with respect to the background screens, since images with and without an object have to be aligned at the pixel level.

Another group of algorithms with explicit background modeling is based on background subtraction. A good review can be found in (Piccardi, 2004; Radke et al., 2005; Parks and Fels, 2008). A background subtraction technique is based on the construction of a background model of a scene at first, followed by the classification of pixels that do not fit this model as foreground pixels. The major drawback of these methods is the requirement of an explicit estimation of the background. This requirement imposes that an update of the background model needs to be done every time the position of the camera is changed which can be difficult for non uniform backgrounds.

An original segmentation technique that is worth mentioning was presented in (Sun et al., 2007). The idea is to use two images: with flash and without flash. It is assumed that the appearance of a foreground object in a ”flash image” is different from that of a ”without flash image”. However, the background remains similar in both images. The main requirement in this method is that the background has to be sufficiently far from the object so that it is not affected by the camera flash. Unfortunately, this condition is not met in our experimental environment.

A more universal way to segment images is to rely on user initialization (Boykov and Jolly, 2001). Here, user input is used to obtain initial information about object and background properties. This information is used to construct a graph and the object segmentation problem is considered as a graph energy minimization. A graph cuts is applied to find the global minimum. In the approach presented in this paper, an energy minimization via graph cuts is also performed to obtain optimal segmentation. However, our goal is to find the initial information required to construct an MRF automatically.

Single image segmentation by graph cut was further extended to automatic silhouette extraction in multiple views in (Campbell et al., 2007; Lee et al., 2007). Although these methods may work well, the usage of explicit object and background color modeling limits the type of objects that can be reconstructed. Another drawback related to color modeling is when the same color belongs to the object and background model. In this case, the result may lead to over- or under-estimation of the silhouette. In our work, we avoid explicit color modeling of an object and background in order to overcome these limitations.



(a) Time-independent step

(b) Time-dependent step

(c) Background likelihood

Figure 1: Background likelihood estimation. (a) - Time-independent step. First the intensity profile is sorted, then $S(x_i)$ is estimated, and finally estimated values are reordered. (b) - Time-dependent step. $S(x_i)$ is estimated on the original intensity profile. (c) - Background likelihood computed as a combination of time-independent and time-dependent steps using equation 5.

3 HYPOTHESIS AND NOTATION

The proposed method is based on the estimation of the background likelihood assuming that the background is unknown but constant and the iterative update of an object likelihood. It is assumed that the camera viewpoint is fixed, and the object changes its position N_T times. In case of multiple lighting conditions, an object is captured N_L times for each frame, one time per source. Note that the proposed method is independent from background modeling, therefore the acquisition process can be repeated multiple times for different camera viewpoints.

The captured image set is organized into a 4D volume. The dimensions of this volume are: U, V, T, and L. U and V are spatial dimensions, T parameterizes object displacement and L represents lighting condition. Thus $I(u,v,t,l)$ is the intensity of a pixel (u,v) at time t under lighting condition l. For notational convenience we define a few shortcuts. $I_L \subset I$ consists of all the images captured under different lighting conditions for a given object position. $I_T \subset I$ is comprised of all the images captured from all the object positions but under fixed lighting. $I_{t,l}$ represents a single image with an object at position t under light source l.

4 BACKGROUND LIKELIHOOD ESTIMATION

In order to estimate background likelihood an "object" and "background" must be defined. A pixel can be called a background pixel if its intensity remains stable for a number of observations among all observations while the object is in motion. This definition follows from the constant background assumption. A pixel whose intensity deviates with respect to its neighbors in time is more likely to represent an object pixel. The definition of an object pixel follows from the fact that during an object motion the orientation of the surface normal of any point on an object changes with respect to the light source or a camera view or both, which is in fact the pixel intensity.

We consider a set of sequential frames as a 3D array and process all subsets of pixels along the time axis. A single subset of pixels form an intensity profile which is defined as: $I_T(u,v) = X = \{I_T(u,v,t_1), I_T(u,v,t_2), \ldots, I_T(u,v,t_i), \ldots I_T(u,v,t_{N_T})\} = \{x_1, x_2, \ldots, x_i, \ldots x_{N_T}\}$ where x_i is the intensity value of a pixel at time i. This profile is depicted by a blue curve in Figure 1.

The core idea of measuring background likelihood is an estimation of the time stability $S(x_i)$ in the intensity profile X. It is measured by estimating the minimum standard deviation around each point. The smaller the deviation, the more stable the point is. Thus, a point with low $S(x_i)$ is most likely to belong to the background. In order to estimate the minimum deviation for a given point $x_i \in X$ a window of size w is slid around it and each time the standard deviation is measured. Among measured values, the minimum has to be found. Formally, the measurement of $S(x_i)$ is defined as follows:

$$S(x_i) = \min_{j \in [i-w+1,i]} \sigma_w(x_j), \qquad (1)$$

where $\sigma_w(x_j)$ is the standard deviation calculated on the subset $\{x_j, x_{j+1}, ..., x_{j+w-1}\}$. $S(x_i)$ describes the constancy of a point x_i in a region with size w.

Since many factors (such as the object's unique geometry, shadows or light inter-reflection in a scene) can affect the intensity of a given pixel, the simple estimation of the stability for each point using equation 1 is not robust enough. Therefore, the estimation of the background likelihood is performed in two steps: ”time-dependent” and ”time-independent”. The necessity of the time-independent step is dictated by the possibility that an object may contain gaps between its parts. In such a case the points inside the intensity profile are mixed between object and background.When the pixels's intensity is analyzed independently of its time order, then one can avoid mixing background and object intensities, as shown in Figure 1(a). The idea of the time-dependent step is to evaluate the property of a point in its original time sequence. It is possible that at some positions, an object point may have the same color intensity as the background. Thus, considering this pixel in its original time sequence order allows a correct estimation of the point deviation as opposed to the time-independent step, see Figure 1(b). The combination of these two steps leads to a reliable estimation of the background likelihood.

The whole algorithm for background likelihood estimation can be summarized as follows:

1. Sort all the points from the intensity profile:
$$X' = sort(X). \qquad (2)$$
2. Time-independent step, see Figure 1(a):
$$S'_g(x'_i) = \min_{j \in [i-w_g+1,i]} \sigma_{w_g}(x'_j). \qquad (3)$$
3. Based on the correspondence between X' and X, reorder S'_g in order to obtain S_g
4. Time-dependent step, see Figure 1(b):
$$S_l(x_i) = \min_{j \in [i-w_l+1,i]} \sigma_{w_l}(x_j). \qquad (4)$$
5. Compute the background likelihood for each point in $x_i \in X$ as follows (Figure 1(c)):
$$P_B(x_i) = \frac{1}{\exp\left(S_g(x_i) + S_l(x_i)\right)}. \qquad (5)$$

Equation 5 is such that it tends to 0 when $S_l + S_g \to \infty$, indicating that the point is inside a varying region and most likely belongs to an object. It tends to 1 when $S_l + S_g \to 0$, meaning that the point is inside a stable region and most likely belongs to the background.

4.1 Space-time Light Volume Fusion

The estimation of background likelihood for space-time volume was described above. If the scene is illuminated uniformly by ambient lighting or only a single light source is used during the acquisition process, then it is enough to use equation 5 to compute the final background likelihood. However, if several directional light sources are exploited, then a fusion process should be applied in order to incorporate information from different light sources. The difficulty of the fusion is caused by contradictory estimations of background likelihoods from different light sources. For example with one light source, some parts of an object can be in the shadow which results in a high value for background likelihood, due to low intensity deviation for such a region. Under another lighting conditions the same part of an object can be well illuminated and thus have a lower background likelihood.

In order to choose an appropriate light source we use a simple but effective rule (a similar approach was used in (Wu et al., 2009) for normal initialization). For a given view and pixel we consider all the images under different lighting conditions, and for each pixel, we find the one that corresponds to a maximum intensity. These lighting condition are used to select the background likelihood for a given pixel:

$$\begin{aligned} max_{ind} &= \arg_l \max(I_L(u,v,l)), \\ P_{B\ final} &= P_B(u,v,max_{ind}). \end{aligned} \qquad (6)$$

4.2 Background to Object Likelihood

Since an estimated background likelihood through equation 6 is just an approximation, the object likelihood cannot rigorously be estimated as $1 - P_B$. Thus we follow the definition of an object pixel (stated in section 4) defined as a high deviation of the intensity profile. The higher the deviation, the closer P_B is to 0. Therefore all the pixels whose background likelihood are close enough to 0 (less than a threshold R) are assigned a value f_1 in order to indicate that there

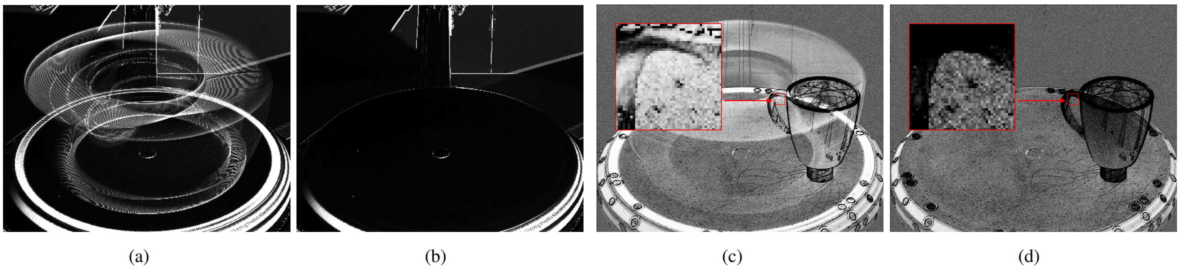


Figure 2: Boundary terms comparison. (a) - γ for diagonal pixel neighbors using raw images: a clear object trace can be seen. (b) - γ for diagonal pixel neighbors excluding object pixels: object influence on gamma disappears. (c) - $B_{p,q}$ with γ from (a), an object trace that is present in γ also affects the boundary term. Some object background boundaries are weakly separated due to that trace. (d) - $B_{p,q}$ with γ from (b), object trace does not appear and a clearer separation between the object and the background for some parts (compare to (c)) is obtained.

is a possibility for an object. The other pixels are assigned the value $f_2 \approx \frac{f_1}{10}$, which indicates that these points are less likely to represent an object. The object likelihood is estimated as follows:

$$P_O = \begin{cases} f_1 & : \quad P_{B\ final} < R \\ f_2 & : \quad otherwise. \end{cases} \qquad (7)$$

5 SEGMENTATION AS AN OPTIMIZATION PROCESS

In the previous section the estimation of prior background and object likelihoods was described. Now the whole segmentation process can be defined. The goal of segmentation is to assign to each pixel p in image $I_{t,l}$ a label m_p which can be the object or the background. Segmentation is performed by minimization of an energy function E through graph cuts (Boykov and Jolly, 2001). Formally,

$$E(M) = \lambda \sum_{p \in I_{t,l}} P(m_p) + \sum_{p,q \in N} B(p,q)[m_p \neq m_q], \quad (8)$$

where $P(m_p)$ is the prior knowledge that each pixel belongs to the object and background; $B(p,q)$ is a boundary term that defines the connection strength between neighboring pixels; M is the set of all labels, each element $m_p, m_q \in M$ can be either background or object with values $\{0,1\}$; λ controls the importance of prior knowledge versus the boundary term ($\lambda \in [0,\infty]$); N is the neighborhood pixel connectivity (in our experiment we use 8-neighbor connectivity).

The boundary term $B(p,q)$ characterizes the relationship between neighboring pixels. If the difference in intensity is small then it is likely that these pixels belong to the same object, therefore they have to be strongly connected. In the case of a large difference in intensity, it is likely that there is an edge and therefore it is probable that these pixels belong to different objects. In such a case $B(p,q)$ should be close to 0 in order to encourage a minimization algorithm to saturate an edge between these points.

Since the object is captured under different lighting conditions, extra information is considered. For example the same point may be in the shadow in one image and may be bright under another light source illumination. This extra data can be used to improve the accuracy of $B(p,q)$. For this purpose, we use the boundary term from (Rother et al., 2004) and modify it in order to incorporate images captured under different light sources (see Figure 2(c)):

$$B(p,q) = \sum_{j}^{N_L} \exp\left(-\frac{||I_{p,i,j} - I_{q,i,j}||^2}{2\gamma_{p,q,j} N_L}\right) \cdot \frac{1}{D(p,q)}, \quad (9)$$

where $I_{p,i,j} = I(u_p, v_p, t_i, l_j)$ and $I_{q,i,j} = I(u_q, v_q, t_i, l_j)$ are intensities for pixel p and q at time t under lighting l_i, $D(p,q)$ is the Euclidean distance between two pixel sites, and $||\cdot||$ is L2-norm. γ is constructed as an expected value over time for each connected pair of pixels. In this way γ is adapted for each viewpoint and lighting condition (see Figure 2(a)):

$$\gamma_{p,q,j} = \sum_{i}^{N_T} \frac{||I_{p,i,j} - I_{q,i,j}||^2}{N_T}. \quad (10)$$

The prior knowledge term $P(m_p)$ in equation 8 defines a preference for pixel p to be object and background:

$$P(m_p) = \begin{cases} P_B & m_p = 0 \quad (background),\ equation\ 6 \\ P_O & m_p = 1 \quad (object),\ equation\ 7. \end{cases} \quad (11)$$

Finally, the energy in equation 8 is optimized through graph cuts and the result of this optimization is a silhouette of an object for each view which is then integrated into the VH.

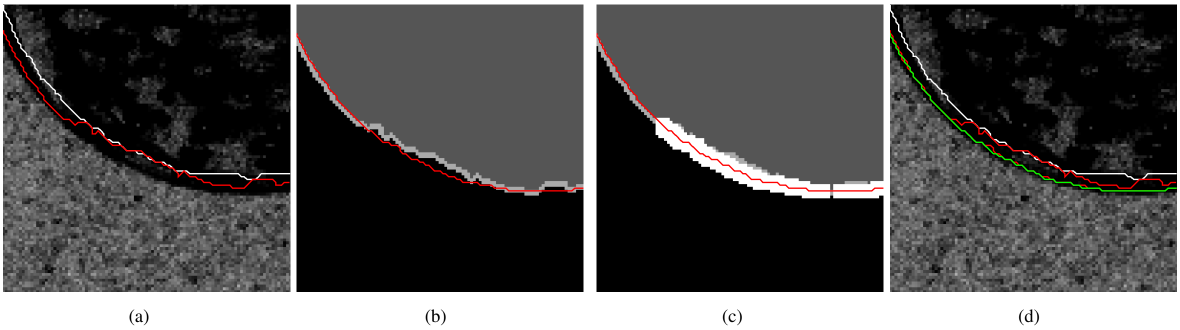


Figure 3: Iterations for updating object likelihood. (a) - boundary term with initial object boundary (white line) and a boundary of the computed silhouette (red line). (b) - updated object likelihood based on equation 13 and the boundary of the new silhouette. (c) - updated object likelihood based on equation 14 and the boundary of the new silhouette. (d) - boundary term with final silhouette boundary (green line), intermediate boundaries (red lines) and initial object boundary (white line).

6 VISUAL HULL REFINEMENT

6.1 Boundary Term Refinement

Having a good approximation of an object shape and its location in each frame allows us to estimate the boundary term more precisely. One of the main parameters of the boundary term is γ. It acts as a threshold: if the difference in intensity between two neighbors is less than γ then the connection between these pixels is strongly penalized. Therefore a clever selection of γ is very crucial for weak edges (when the difference between neighbors is quite small). Thus, it is important to estimate γ as precisely as possible to obtain pure connectivity of background neighboring pixels. Therefore the following procedure was adopted: the VH is projected onto each frame and pixels that belong to the silhouette are excluded from the the calculation of γ. This exclusion does not eliminate all the shading effects such as shadows, inter-reflections and color bleeding but their effect is almost negligible and is even reduced by signal averaging over time.

The result is that γ is computed almost only between non object pixels, which is in some way similar to computing γ on the background image (without an object):

$$\gamma_{p,q,j} = \sum_{i}^{N_T} \frac{||I_{p,i,j} - I_{q,i,j}||^2}{N_T}, p,q \notin Pr_i^{-1}(H), \quad (12)$$

where $Pr_i^{-1}(H)$ is a silhouette of the projected VH H on frame i. The result of such an update is shown in Figure 2(b): the influence of an object's motion on γ almost disappears with the result that a pure background connectivity information between neighboring pixels is estimated.

Substituting γ computed with equation 12 in the boundary term in equation 9 produces more accurate results, see Figure 2(d). In Figure 2(c) the boundary term computed with the initial γ formulation is shown. As it can be seen, some boundary parts between the cup handle and the background are weakly separated due to the presence of the object's motion trace in γ, see Figure 2(a). However when the object motion is eliminated from γ (Figure 2(b)) a clearer separation is obtained. One of the issues with the new formulation of γ in equation 12 is that the resulting boundary term becomes more sensitive to image changes. It can be seen that much more neighboring weights inside an object receive low penalty compared to the initial formulation of γ (see Figures 2(c) and 2(d)). Nevertheless, it is not critical since edges between object and background are detected more accurately and non-zero object likelihood covers almost the entire object. Therefore, only edges close to the object boundary play an important role when maxflow is computed. Note that by computing an adaptive γ for each neighboring pixel connection, most of the background edges are eliminated. In our scene a non uniform background with many edges can be observed, nevertheless almost all the background edges do not appear in boundary term (see Figure 2(d)). The formulation of this term is one of the contributions of this work.

6.2 Iterative Refinement of Object Likelihood

One source of inaccuracy is the strong edge on the object near the boundary. It is possible that the boundary of the computed silhouette can pass through such strong internal object edges. Therefore we try to find such places and push the boundary out in order to bypass these internal edges. For that reason we apply the following strategy: first we try to push the boundary of the obtained silhouette. If some parts of the boundary move, then we adjust these parts by searching for another strong edge nearby.

As an initial step, all the points that belong to the silhouette are assigned weight w, points located no further than T_1 to the closest point of the silhouette are assigned weight $2*w$

$$P_{O_1}(x_i) = \begin{cases} w & : \ x_i \in S \\ 2*w & : \ dist(x_i,S) < T_1, x_i \notin S \\ 0 & : \ otherwise \end{cases} \tag{13}$$

Such an update of the object likelihood allows the potential identification of internal object edges that were accepted as the object boundary during the initial calculation of maxflow.

In the second step all the points of a computed silhouette that coincide with the zero region of P_{O_1} or with its boundary form a set C. This set represents points that are close to or belong to an internal object edge. We want to push the silhouette boundary that is inside C to overcome internal edges and move it toward the real object boundary. Therefore the object likelihood is updated as follows:

$$P_{O_2}(x_i) = \begin{cases} 3*w & : \ dist(x_i,C) < T_2, \\ P_{O_1}(x_i) & : \ otherwise \end{cases} \tag{14}$$

We continue to update the object likelihood using equation 14 and maxflow calculation until set $C \neq 0$ or until the maximum number of iterations is reached.

All these steps are illustrated in Figure 3. In Figure 3(a) the boundary term with the initial object border (white line) and the resulting silhouette border (red line) are depicted. As it can be seen, the boundary of the silhouette goes through the edge inside the object. A new object likelihood is constructed based on equation 13, see Figure 3(b) and the boundary of the resulting silhouette is depicted by the red line. It can be seen that an internal object edge was crossed. Since the resulting silhouette is not totally inside the non-zero region of P_{O_1}, set C is not empty. Therefore, the object likelihood is updated again based on equation 14 (see Figure 3(c)). Finally, the resulting boundary (red line) is completely inside the non-zero region of P_{O_2} and therefore, C is empty. The final silhouette boundary (thick red line) with the boundary term is depicted in Figure 3(d). The part of the initial boundary that was inside an object was pushed towards the object boundary and the rest of the boundary that was close to the true object-background edge was just slightly adjusted.

Note that when two object parts are separated by the background and the distance between the closest object points is less or equal to T_2, such regions are joined together in the resulting silhouette. This problem is addressed by the final step of the algorithm.

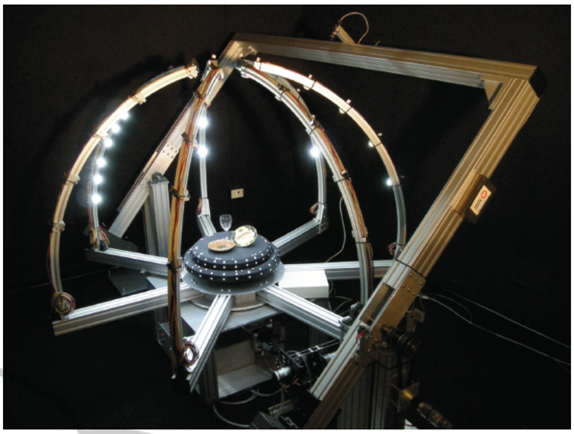


Figure 4: Acquisition system. It consist of a turntable, a lighting system and a camera system.

6.3 Visual Hull Completion

Finally, in order to enforce silhouette boundary smoothness and coherency between frames, a graph cuts on a set of sequential frames is performed. Several consecutive frames are considered together and treated as a 3D array. A new graph is constructed in a way similar to what was done previously for each individual frame except for two differences.

As a first difference, the object likelihood is taken from the last step of the iterative algorithm described in section 6.2. All the values that belong to the silhouette of the projected VH are taken from P_{O_2}, the rest are set to zero.

$$P_{O_3}(x_i) = \begin{cases} 0 & : \ x_i \notin Pr_i^{-1}(H), \\ P_{O_2}(x_i) & : \ otherwise. \end{cases} \tag{15}$$

In using this construction of the object likelihood one can overcome the problem of merging nearby object areas mentioned in section 6.2. Since points located outside of the projected VH are set to 0, a strong object enforcement is eliminated for inter-object areas while the rest of the object likelihood remains the same.

A second difference is that sequential frames have to be connected together by inter-frame arcs in the graph. Based on the object motion between two frames, we can identify which graph nodes must be connected between frames. Using the VH and calibration information for each frame allows the most common object motion directions to be found between two frames. VH voxels are first projected in each frame and then all the projected voxels falling into the boundary of the silhouette at least in one frame are used to form a set of directions:

$$D = Pr_i^{-1}(H(v_i)) - Pr_{i+1}^{-1}(H(v_i)), \forall v_i \in H. \tag{16}$$

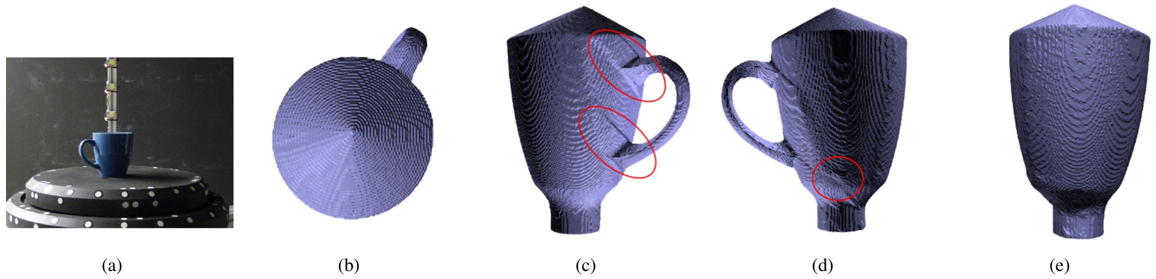

(a) (b) (c) (d) (e)

Figure 5: VH of a cup. (a) - an image of cup. (b)-(e) - the VH of the cup from different viewpoints. In (b) a small region near the cup handle goes beyond since the cup handle hides this part from direct observation in several views. In (c) - a small bump can be observed due to target merging with the cup silhouette in some views.

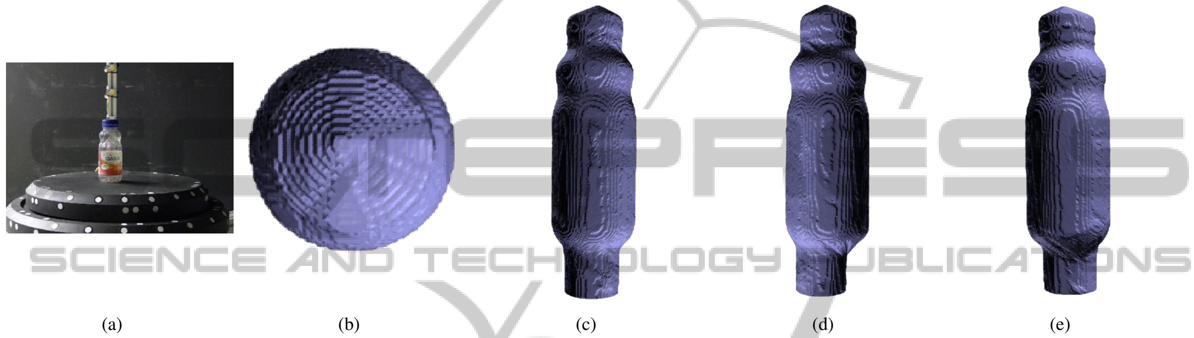

(a) (b) (c) (d) (e)

Figure 6: VH of a plastic bottle. (a) - an image of a plastic bottle. (b)-(e) - the VH of the the bottle from different viewpoints.

The set of directions *D* may contain a large number of different directions. Therefore, only the most common directions are selected (typically between 8 and 15) to connect nodes between frames. The weight for each inter-frame arc is computed using equation 9. The background likelihood term and the boundary term are constructed the same way as explained previously.

7 EXPERIMENTAL RESULTS

The experiments for validating the approach are performed with a roboticized system of our design, which allows the position of a turntable, the camera position on a hemisphere above the turntable and the lighting condition to be controlled by a computer, the setup is shown in figure 4.The background behind an object is not uniform, it consists of: a wall, different parts of the setup and a turntable with white calibration discs. The camera viewpoint is not constant and can be easily changed which leads to a complete change of the observed background. The proposed approach was tested on several objects with complex surface properties. In a typical experiment, an object is rotated 360 times by 1 degree increments and a grayscale image is captured under 30 different lighting conditions. In cases when the object surface shows specular properties it may reflect light to an area near its base and thereby violate the assumption of the constant background area near this location. By resting the object on a small pedestal on a turntable, this effect is reduced significantly and therefore can be neglected.

Figure 5 shows the VH of a cup. The cup has a smooth conical shape, is made from ceramic and its surface is covered by uniform glossy paint which causes specular reflections and non constant appearance to be observed during image acquisition. Another complication is the difficulty of finding distinctive features on the object for multi-view matching. Such object properties highly complicate the reconstruction of the geometry for feature-based methods. A few traces (enclosed in red ellipses) near the cup handle can be seen (Figure 5(c)). They appear due to the fact that this area is hidden by the cup handle from direct camera observation in several consecutive views. Also a small bump can be observed near the bottom of the cup base (Figure 5(d)). It is caused by some circular targets on the turntable treated as part of the silhouette since they match the definition of an object. Despite these small errors, the proposed method was able to reconstruct the cup correctly.

In Figure 6 the results of the segmentation of a

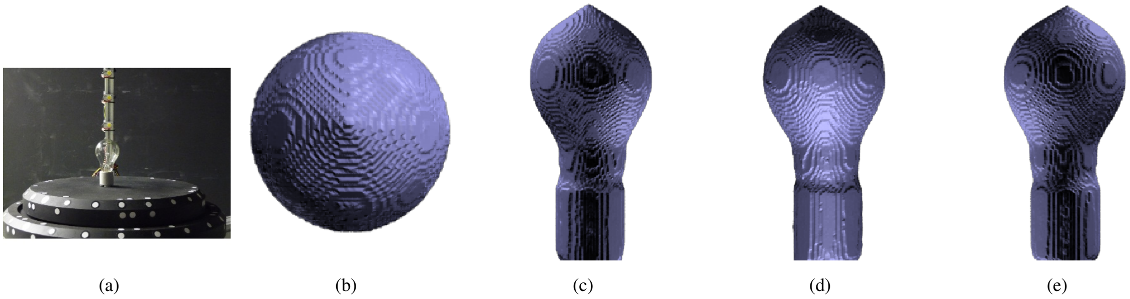


Figure 7: VH of a light bulb. (a) - an image of a light bulb. (b)-(e) - the VH of a light bulb from different viewpoints.

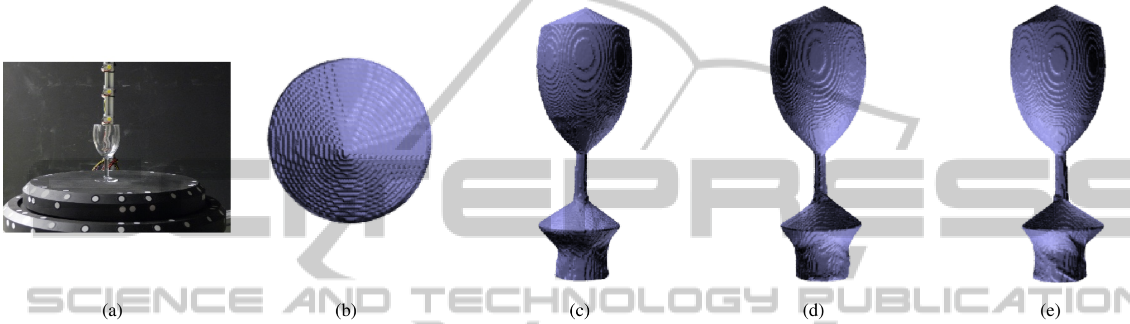


Figure 8: VH of a wine glass. (a) - an image of a wine glass. (b)-(e) - the VH of a wine glass from different viewpoints.

plastic juice bottle are presented. Geometry reconstruction for such an object is a challenging task for several reasons. An area close to the boundary edges of the bottle is transparent, therefore the intensity of this part coincides often with the intensity of the background. The intensity of the bottle lid is similar to the surrounding background in some frames, which also complicates object-background separation. Notwithstanding these conditions, the VH of the bottle is reconstructed accurately.

Finally the algorithm was tested with fully transparent objects: a light bulb and a wine glass. Due to the transparency, it is practically useless to try to estimate distribution of object colors or to search for distinctive object features, as only the properties of the scene located behind the object will be observed. Another complication with a transparent object is that during its motion, a different background is observed, which makes it difficult to estimate a consistent feature and color model between several views. As it can be seen in Figures 7 and 8, the body of the light bulb and the wine glass are transparent and the background is visible through them. Since our approach is not based on object color features modeling, it is possible to obtain a reliable reconstruction of the geometry of both the bulb and the wine glass.

8 CONCLUSIONS

In this paper an approach for the reconstruction of the Visual Hull of an object with complex photometric properties was described. The proposed approach is based on two principles: modeling scene background based on signal stability which is independent of camera viewpoint and then iterative updating the object likelihood to refine the estimated silhouette boundary accurately.

The advantage of the proposed approach is that instead of attempting to model the object color space or matching object features, the evolution of pixel intensity over time is analyzed. Such an analysis avoids the use of standard objects property, such as color and edges and allows the VH to be reconstructed for a wide range of objects with different shapes and reflective properties without any prior knowledge. We show that the proposed method is capable of dealing with objects with complex reflectance properties such as textureless objects or completely transparent ones. The requirement for handling a wide variety objects with completely different photometric properties is that a dense set of images is required for the construction of the Visual Hull. As a future work, we plan to use photometric information for estimating object reflectance properties and fuse this information with the VH to obtain complete object description.

REFERENCES

Baumgart, B. G. (1974). *Geometric modeling for computer vision*. PhD thesis, Stanford, CA, USA.

Boykov, Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Eighth IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 105 –112.

Campbell, N., Vogiatzis, G., Hernndez, C., and Cipolla, R. (2007). Automatic 3d object segmentation in multiple views using volumetric graph-cuts. In *British Machine Vision Conference*, volume 1, pages 530–539.

Jagers, M., Birkbeck, N., and Cobzas, D. (2008). A three-tier hierarchical model for capturing and rendering of 3d geometry and appearance from 2d images. In *International Symposium on 3-D Data Processing, Visualization, and Transmission (3DPVT)*.

Lee, W., Woo, W., and Boyer, E. (2007). Identifying foreground from multiple images. In *Eighth Asian conference on Computer vision (ACCV)*, pages 580–589.

Matusik, W., Pfister, H., Ngan, A., Beardsley, P., Ziegler, R., and McMillan, L. (2002). Image-based 3d photography using opacity hulls. *ACM Transactions on Graphics*, 21(3):427–437.

Parks, D. H. and Fels, S. S. (2008). Evaluation of background subtraction algorithms with post-processing. In *International Conference on Advanced Video and Signal Based Surveillance*, pages 192–199.

Piccardi, M. (2004). Background subtraction techniques: a review. In *International Conference on Systems, Man & Cybernetics (SMC)*, pages 3099–3104.

Radke, R. J., Andra, S., Al-Kofahi, O., and Roysam, B. (2005). Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307.

Rother, C., Kolmogorov, V., and Blake, A. (2004). "grab-cut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314.

Smith, A. R. and Blinn, J. F. (1996). Blue screen matting. In *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 259–268.

Snow, D., Viola, P., and Zabih, R. (2000). Exact voxel occupancy with graph cuts. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:1345.

Sun, J., Kang, S. B., Xu, Z., Tang, X., and Shum, H.-Y. (2007). Flash cut: Foreground extraction with flash and no-flash image pairs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wu, C., Liu, Y., Ji, X., and Dai, Q. (2009). Multi-view reconstruction under varying illumination conditions. In *Proceedings of the IEEE international conference on Multimedia and Expo*, pages 930–933.

Zongker, D. E., Werner, D. M., Curless, B., and Salesin, D. H. (1999). Environment matting and compositing. In *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 205–214.