

# Shadow Multiplexing for Real-Time Silhouette Extraction

Tom Cuypers

Yannick Francken

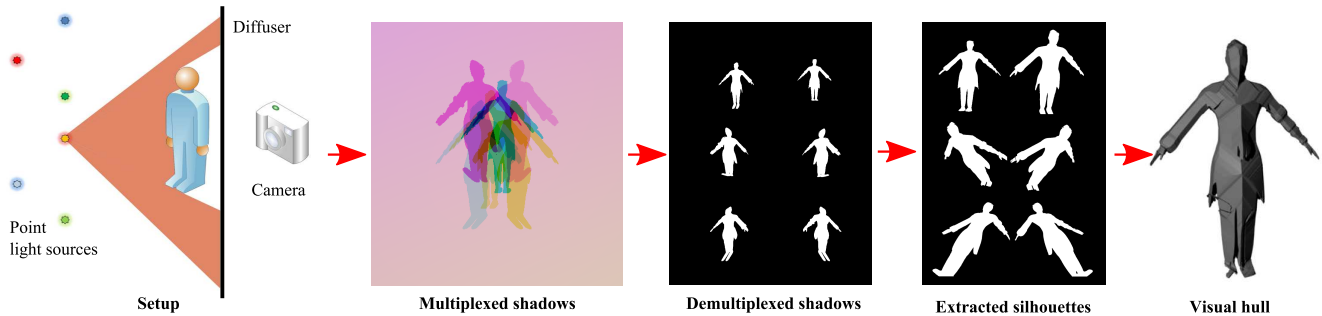
Johannes Taelman

Philippe Bekaert

Hasselt University - tUL - IBBT

Expertise Centre for Digital Media, Belgium

{firstname.lastname}@uhasselt.be



**Method overview.** The setup consists of multiple colored point light sources, a diffuser and a digital camera. For each light source a silhouette is extracted from the captured shadows of the scene. These silhouettes are used for visual hull reconstruction as well as image based collision detection.

## Abstract

*In this work we propose a real-time implementation for efficient extraction of multi-viewpoint silhouettes using a single camera. The method is based on our previously presented proof-of-concept shadow multiplexing method. We replace the cameras of a typical multi-camera setup with colored light sources and capture the multiplexed shadows. Because we only use a single camera, our setup is much cheaper than a classical setup, no camera synchronization is required, and less data has to be captured and processed. In addition, silhouette extraction is simple as we are segmenting the shadows instead of the texture of objects and background. Demultiplexing runs at 40 fps on current graphics hardware. Therefore this technique is suitable for real-time applications such as collision detection. We evaluate our method on both a real and a virtual setup, and show that our technique works for a large variety of objects and materials.*

## 1. Introduction

Many virtual reality applications require a natural interaction between the real and the virtual world. An intuitive way to achieve this is to create a 3D model of the human actor, perform collision detection and provide an appropri-

ate response in the virtual and/or real scene. As this process needs to operate in real-time without being visually rendered, an approximate model often suffices. A traditional approximation for real-time modeling is the shape-from-silhouette technique [2, 17]. In a multi-camera setup, silhouettes are obtained by performing background subtraction on every image. They are then combined to reconstruct the original volume. As an increasing number of silhouettes often results in a better reconstruction, a large amount of calibrated and synchronized cameras is needed to produce accurate results, making this an expensive and complex setup. The vast amount of data generated by these cameras makes it ill-suited for real-time full-resolution processing on a single computer.

Therefore we proposed a technique that extracts multiple silhouettes from different views in a single image, by replacing the cameras with colored light sources and capturing the cast shadows on a diffuser [6]. This method is related to shadow carving [22] and coplanar shadowgrams [27], but instead of extracting a set of silhouettes from subsequent images, we obtain multiple silhouettes from a single image. In this way, our approach is very similar to the recent work of Lanman et al. [16], although their method obtains silhouettes from a light field by using a specialized dense lens array, where we achieve our silhouettes from segmenting colors on a common diffuser. Our setup

is consequently cheaper and we do not need to sacrifice any camera resolution when light sources are added. We are only bound to the color resolution of the camera, which in practice allows for easily extracting six silhouettes from a single shot. This has proven to be sufficient for the purpose of collision detection [8].

The colors in shadow regions are mainly influenced by the colors of the light sources, and very little by the reflection properties of the object. Therefore background subtraction is more robust than standard texture-based background extraction used by traditional visual hull methods. Furthermore, since we only require a single camera, it is possible to use *one* high quality camera instead of a number of low quality cameras for the same price. This single camera will record the backside of the diffuser. The obtained image is used to extract the different full-resolution silhouettes as if they were captured by virtual cameras from the location of the point light sources. Because only one camera is used, there is no need for synchronization and only a single image needs to be processed. The acquired silhouettes are directly used for efficient collision detection in image space, skipping any visual hull reconstruction or image transformations. This results in a significant speedup compared to performing collision detection with traditional visual hull methods.

This work is a real-time implementation of our previously presented proof-of-concept poster about shadow multiplexing [6], applied to collision detection. In addition, a mathematical foundation, comparison with respect to the state-of-the-art and a more extensive discussion of the results are presented in this extended version.

## 2. Related Work

A lot of research has been done in interaction between real actors and virtual scenes [1, 11, 8]. In order to achieve this goal, some researchers reconstruct a 3D model of the actor, more in particular a visual hull [11, 7]. Others try to avoid creating this virtual structure and perform image based interaction [8, 18]. For these techniques, silhouette extraction is an important step and this will be our primary focus.

Although a visual hull is well suited for real-time interaction, it comes with some disadvantages. First of all, it often requires an expensive setup because many cameras are required to obtain a detailed model of the actor. Alternatively, cameras can sometimes be replaced by controllable light sources for reconstructing 3D shapes, as shown in miscellaneous shape acquisition methods [23, 9, 13, 26]. Similarly, in this work we will try to replace as many cameras as possible by much cheaper light sources. Our setup consists of only *one* camera since the required functionality of other cameras is replaced by a combination of power LEDs and a diffuser as shown on Figure 1.

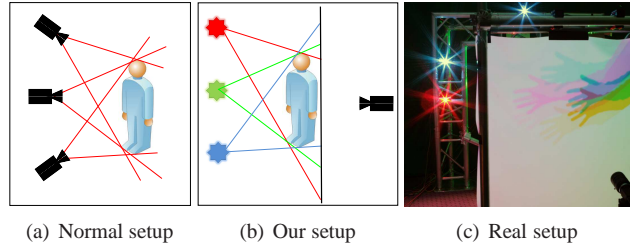


Figure 1. (a) Typical setup to create a visual hull (b) The cameras are replaced by light sources (c) Our setup consisting of multiple colored power LEDs, a diffuser and a camera.

Another problem is the large amount of data that is produced by the multiple cameras, all of which needs to be processed. It is therefore common for multiple computers to be used simultaneously for real-time visual hull reconstruction [11, 8, 4]. Others obtain a 3D model using single shot techniques [14, 15]. This way no synchronization is required between the cameras. Kawasaki et al. [15] present a single shot structured light method which provides a disparity map for one viewpoint, while we require multiple views to provide an all around collision detection. Huang and Lai [14] proposed a single camera setup for visual hull reconstruction consisting of mirrors to multiplex multiple viewpoints into a single image. However, these viewpoint images are not allowed to overlap, limiting the resolution of the silhouettes. Our method is also a single shot technique and therefore does not require any synchronization. It can run in real-time on a single computer and has no resolution limits.

A third disadvantage of visual hull reconstruction is the necessity for background subtraction as a preprocessing step to obtain silhouettes for each camera. There are numerous methods for background subtraction [19, 5], but for high performance a static known background is usually assumed [25]. We provide an easy and fast silhouette extraction for a large variety of opaque materials by observing the cast shadows.

We choose to use the shadows of the subject because they can be obtained with only a single camera. In contrast to other methods that observe shadows for shape reconstruction [27, 22, 12], the shadows are multiplexed so that more than one can be extracted from a single image. Lanman et al. [16] also provide a method of extracting multiple shadows from the same image, but they require expensive and specialized hardware. Their method is image resolution bound as the camera resolution has to be divided by the number of light sources in the scene. Our method is able to extract full resolution silhouettes, but is color resolution bound instead. This means that in our practical setup up to six instead of tens of silhouettes can be extracted. However, we provide a simpler and cheaper setup as we only need

a limited number of silhouettes for our collision detection application.

Summarized, the advantages of our method are:

- Cheap setup: only a single camera and a few low cost power LED light sources are required
- Simple and robust silhouette extraction by segmenting shadows instead of background subtraction
- No camera synchronization
- Full camera resolution available
- Limited amount of data to be captured and processed

### 3. Shadow Demultiplexing

Our method replaces multiple cameras by simple light sources, as shown in Figure 1. Each light source casts a shadow on the diffuser behind the subject. We will show that multiple shadows can be obtained in a single shot. In the following section we will describe several multiplexing methods. In the next section we will go into more detail about demultiplexing shadows, followed by an optimal light source configuration.

#### 3.1. Types of Multiplexing

In a scene with multiple point light sources, we have multiple shadows. These shadows can be obtained by a single image and there are several methods to make it possible to extract the individual shadows. This is called shadow demultiplexing. Schechner [24] also presents a method to multiplex different light sources into a single image with the intention to reduce camera noise, not reducing the number of captured images.

We distinguish between four commonly known types of shadow multiplexing techniques (Figure 2):

1. *Time Multiplexing*: at each time only a few light sources are turned on, making it easier to distinguish the different shadows
2. *Space Multiplexing*: the position of the shadow defines its corresponding light source. E.g. when shadows do not overlap, the leftmost shadow is coming from the rightmost light source
3. *Color Multiplexing*: when using a red, green and blue light source, the different shadows are extracted from an RGB image
4. *Intensity Multiplexing*: every light source has a different intensity casting mixed shadow intensities

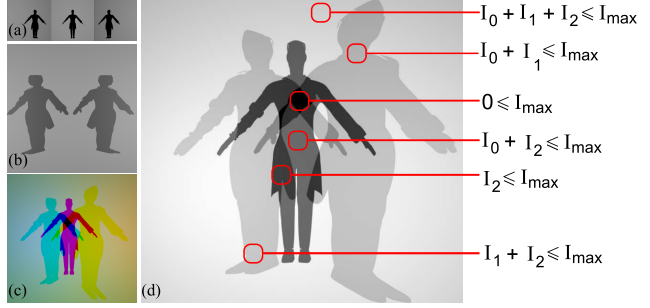


Figure 2. Shadow multiplexing types (a) time (b) space (c) color (d) intensity

Since we want a single shot recording without any loss of resolution, we prefer to combine color and intensity multiplexing simultaneously. As color multiplexing can trivially be achieved by treating the three RGB channels separately, we will mainly focus on intensity demultiplexing within a single color channel.

#### 3.2. Intensity Multiplexing

In order to optimally demultiplex shadows cast under multiple point light source illumination, we will build a model based on the assumptions we make. We focus on light sources all emitting light of the same frequency but having a different intensity  $E_\omega$ . For a certain location  $x$  on the diffuser, the intensity  $I_C$  will be the integral of incoming light over the hemisphere  $\Omega$ , taking into account which directions  $\omega$  are occluded ( $C_\omega = 0$ ) or not ( $C_\omega = 1$ ):

$$I_C = \int_{\Omega} C_\omega \frac{1}{d_\omega^2} \rho E_\omega \cos(\omega, n) d\omega \quad (1)$$

where  $\frac{1}{d_\omega^2}$  is the quadratic attenuation due to the light source distance  $d_\omega$ ,  $\rho$  the diffuse albedo,  $E_\omega$  the emitted intensity and  $\cos(\omega, n)$  the attenuation due to the incident angle at location  $x$ .

Since we assume there is no stray light and no interreflections, hard shadows are directly formed by the  $N$  available point light sources. This allows us to discretize the integral by removing directions  $\omega$  where no light sources  $i$  are placed:

$$I_C = \sum_{i=1}^N C_i \frac{1}{d_i^2} \rho E_i \cos(\omega_i, n) \quad (2)$$

In a static setup, the reflection and illumination parameters can be grouped in a single parameter  $I_i$  for every light source  $i$ :

$$I_C = \sum_{i=1}^N C_i I_i = C \cdot I \quad (3)$$

The goal is now to find the occlusion vector  $C$  for each pixel from the captured intensity  $I_C$ . If the parameters  $I_i$  are carefully chosen, for every possible occlusion vector a unique  $I_C$  value is obtained. Hence, given a recorded intensity  $I_C$ , we can find the occluded light sources  $C$ .

In order to make the parameter selection possible,  $I_i$  has to be almost the same for every surface location. Therefore we must get rid of the location dependent fall-off due to the cosine and light attenuation. This is achieved by placing the light sources at a far enough distance from the diffuser. In section 3.3 we will propose a technique to optimally choose the parameters  $I_i$ .

Notice that we demultiplex the shadows on a per pixel basis, making this method suitable for parallel processing. This will be illustrated with the results of our GPU implementation in section 5. In this implementation, intensities are independently multiplexed for each of the three available color channels (red, green and blue).

### 3.3. Optimal Light Source Configuration

Since the silhouette quality strongly depends on the configuration of the light sources, we will have a look at how to position them and how to determine the optimal intensities.

#### 3.3.1 Positioning

In order to obtain a high quality visual hull, light sources have to be placed in a wide baseline fashion. However, there are extra constraints we have to take into account as we want to get rid of location dependent light fall-off due to the incident angle and quadratic light attenuation. We achieve this by placing the light sources at a sufficiently large distance away from the screen. Consequently the incident light angle will become almost equal for different locations on the diffuser. The same applies for the light attenuation where the difference in distance will be of less importance when all the light sources are located “far” away from the surface. As a precaution in cases where the requirements cannot be sufficiently met, we place light sources of a similar frequency but different intensity in each other’s neighborhood. This increases the robustness in the demultiplexing phase as the light sources will experience a similar fall-off.

#### 3.3.2 Intensities

Finally, assuming that we can easily separate different color channels, we have to define which intensity each light source should have. Some restrictions need to be imposed for practical purposes. In order to minimize the influence of noise, each light source intensity  $I_i$  should have a minimum value  $I_{min}$ .

$$\forall_i : I_i \geq I_{min} \quad (4)$$

The sum of the light sources is not allowed to saturate, to

$N$	$I_{min}$	$I_{max}$	$I$	$\Delta I$
2	30	255	{84, 168}	84
2	50	255	{84, 168}	84
3	30	255	{36, 72, 144}	36
3	50	255	{56, 84, 112}	28
4	30	255	{30, 44, 58, 116}	14
4	50	255	{50, 57, 64, 78}	7

Table 1. Optimal intensity values  $I$  for intensity shadow multiplexing for  $N$  lightsources with a minimum ( $I_{min}$ ) and maximum ( $I_{max}$ ) intensity.

avoid loss of intensity information, which causes ambiguous demultiplexing (Figure 2(c)). For  $N$  light sources we get:

$$\sum_{i=1}^N I_i \leq I_{max} \quad (5)$$

Eventually we want the difference between every two configurations  $a$  and  $b$  to be as large as possible, with a configuration the combination of lights that illuminate a certain point. We can formulate this constraint as

$$\forall_a \forall_{b \neq a} |(C^a \cdot I) - (C^b \cdot I)| \geq \Delta I \quad (6)$$

where  $C_i^a$  can be 1 or 0 meaning that for a certain configuration  $C^a$ , light source  $i$  is occluded or not. We can calculate optimal values for  $\{I_1 \dots I_N\}$  by maximizing  $\Delta I$  for the constraints (4)-(6). Optimal intensity values are presented in table 1.

## 4. Setup

In this section we will explain the details of the hardware involved as well as how to calibrate it.

### 4.1. Hardware

Our setup consists of several point light sources for which we use very bright LEDs. Each LED emits a different color and intensity to make it possible to demultiplex. We constructed this setup with a PGR Grasshopper camera [21], a white back-projection diffuser and several Osram Diamond Dragon LEDs [20]. The LEDs are placed at an approximate distance of 2 meters away from the center of the diffuser. These are single-chip high brightness LEDs with a wide angle radiation characteristic. Figure 1(c) shows an image of our setup. Table 1 indicates that in a single color channel even 3 or 4 intensities can theoretically be demultiplexed with an 8 bit camera. Unfortunately, in practice we are bound to 3 intensities for the red color channel and 2 for the green and blue channels. This is due to the fact that a CCD camera has a strong green response for blue LEDs, as will be explained in section 5.1.4.

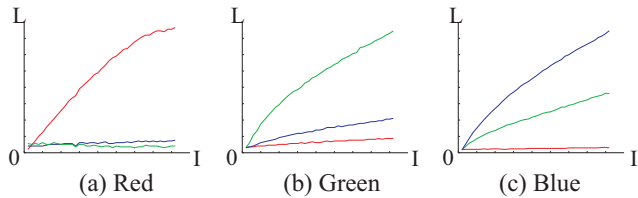


Figure 3. Relationship between the luminance and current of a (a) red, (b) green and (c) blue LED. Notice the strong green response of the CCD camera for the blue LED.

## 4.2. Calibration

In order to provide our algorithm with the light properties of the setup and the relative positions of the hardware components, we will now cover radiometric and geometric calibration.

### 4.2.1 Radiometric

Each LED is connected to a controllable power source. By adjusting the current sent through the LED, more light will be emitted. As can be seen in Figure 3, the luminance will not respond linearly in function of the current. This plot can be measured or found in the characteristics provided by the LED manufacturer. In practice this means that the current required for getting a certain intensity value can simply be found in a lookup table containing the LED characteristics.

### 4.2.2 Geometric

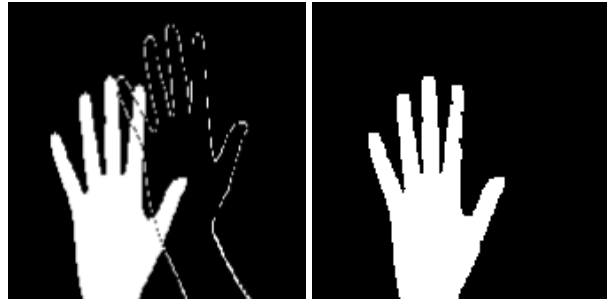
The geometric calibration of the setup consist of scanning the 3D structure of the diffuser and defining the location of each point light source. We assumed that our diffuser is planar, simplifying the calibration. We calibrate it by attaching a checkerboard pattern and using a standard calibration method [3, 10]. More in general all kinds of 3D scanning techniques, such as structured light [23] can be used for scanning the diffuser.

Finding the location of each light is achieved using the calibration proposed by Yamazaki et al. [27]. It analyzes the ellipsoidal shadows cast from occluding spheres.

## 5. Results

In this section we will show results we obtained and the problems we encountered. We will also demonstrate two applications where the extracted silhouettes are used.

In figure 7 we show a couple of extracted silhouettes. Herefore we use two red, two green and two blue LEDs with different intensity. These results indicate that this technique is suitable for different types of materials like low albedo or specular surfaces.



(a) Demultiplexed shadow (b) Postprocessing

Figure 4. Captured images contain soft shadows: (a) Straight forward shadow demultiplexing method (b) Figure (a) after performing erosion and dilation

## 5.1. Problems

The straightforward method for demultiplexing can sometimes introduce some errors and artefacts which we show can be reduced.

### 5.1.1 Soft Borders

A point on the diffuser does not only receive light from each light source, but also from the neighbouring illuminated points. The diffusion due to scattering within the diffuser results in a smoothed transition at the borders of the shadows. These soft borders sometimes create a ghost silhouette as shown on Figure 4(a). Most of the time they have a size of only a few pixels. Therefore they can easily be removed by performing morphological operations such as erosion and dilation as shown on Figure 4(b).

Soft borders can also occur due to motion blur. We can minimize these by lowering the shutter time of our camera. Therefore we require brighter LEDs. LED strobing is a common method for obtaining the same intensity with a lower shutter time reducing motion blur. However, the OSRAM Diamond Dragon LEDs do not allow significant extra light output in pulse operation, so this technique will require a different selection of LEDs.

### 5.1.2 Camera noise

Shadow demultiplexing is a local algorithm meaning that camera noise can result in faulty results at some pixels. Even though this noise is already minimal, it will even be further reduced by the erosion operation.

### 5.1.3 Object Material

Our method works well for sufficiently opaque materials such as high and low albedo diffusers, glossy and specular surfaces, and many translucent materials. However, as is the case for standard background subtraction, this method

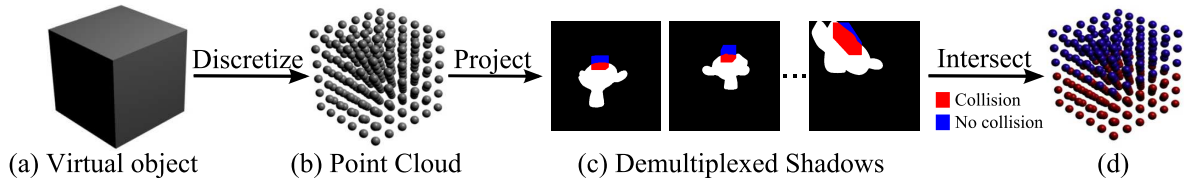


Figure 5. Collision detection overview (a) input, a virtual object we want to check collision with (b) the virtual object converted into a point cloud (c) the point cloud is projected onto the demultiplexed shadows of the real object (d) red points are inside every silhouette after projection, while blue points are projected outside at least one silhouette.

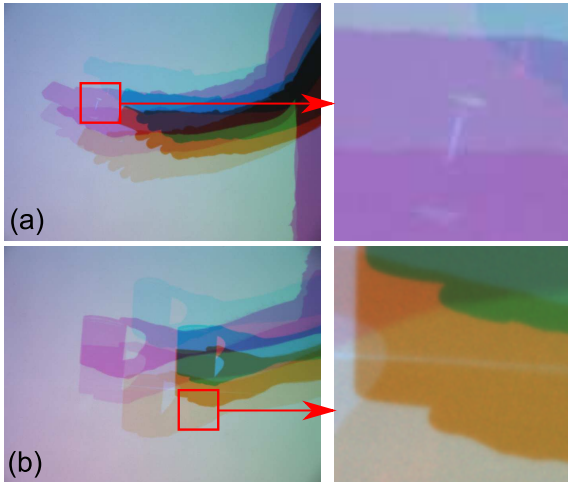


Figure 6. Artifacts due to (a) specular reflections and (b) refractions

will break down for mainly transparent materials. Occasionally small artefacts appear when highly specular materials reflect their light directly toward the surface as shown in Figure 6.

### 5.1.4 Blue-Green Response Overlap

Because we do not use a multispectral camera, the number of multiplexed blue and green LED intensities was limited by two per light source. This is due to the large response overlap between green and blue on CCD cameras. This is illustrated in Figure 3.

## 5.2. Timings

The method is implemented on an NVidia GForce 8800 GTX. We store up to 4 silhouettes into a single texture, reducing required memory storage. It also takes advantage of the scalar processors. The number of render passes is also reduced by 4 and the erosion and dilation can be calculated simultaneously. For input images containing 6 silhouettes with a resolution of  $640 \times 480$  we achieve a framerate of 40 fps.

## 5.3. Applications

In this section we will demonstrate the usefulness of our method for interactive collision detection. The first application shows how we can reconstruct a visual hull from these silhouettes, which can then be used for collision detection with virtual objects. As a second application we show how the silhouettes can be used for collision detection in image space.

### 5.3.1 Visual Hull Reconstruction

The visual hull of the object can be reconstructed using classical shape-from-silhouette techniques. Because these methods usually require silhouettes from the viewpoint of multiple cameras we have to perform a transformation on the demultiplexed shadow. Since the setup is calibrated we can use projective texture mapping for an efficient transformation. We model the diffuser on which we project our multiplexed shadows. Then we render it from the position of the light source. Because the shadows are already demultiplexed on the GPU, this step can easily and efficiently be added. An overview of this method is shown in the teaser image.

### 5.3.2 Image-Based Collision Detection

Collision detection can also be done directly in image space. Therefore we divide our virtual objects into small voxels and project each voxel onto the input image. We can then determine if this voxel is projected inside or outside the real object's silhouette. When a projection falls inside every silhouette, we assume a collision has occurred. This way collision response can be handled directly in image space. This is similar to the work of De Decker et al. [8]. An overview of this method is shown in Figure 5.

## 6. Conclusion

We proposed a technique to efficiently obtain multiple full resolution silhouettes of an object from multiplexed shadows using a single camera, allowing to construct a visual hull or provide image based collision detection in real-time on a single pc. Because only one camera is required,

no synchronization is needed and less data has to be obtained and processed, increasing the performance and accuracy. Furthermore, easy and robust background subtraction is achieved for a variety of object materials.

## 7. Future Work

Currently we manually control the intensities of our LEDs. We are working on a computer-controlled LED driver to automate LED intensity calibration, providing a more accurately tuned configuration.

In order to increase the number of silhouettes in our prototype, we are looking to replace the RGB camera with a multispectral camera since some of the method's potential is lost due to the considerable blue-green response overlap of RGB cameras.

## Acknowledgements

The authors (Cuypers Tom and Philippe Bekaert) acknowledge financial support by the European Commission (FP7 IP "2020 3D media"). We gratefully express our gratitude to the European Regional Development Fund (ERDF), the Flemish Government and the Flemish Interdisciplinary institute for Broadband Technology (IBBT), which are funding part of the research at the Expertise Centre for Digital Media. We also would like to thank our colleagues for their time and effort in this work.

## References

- [1] S. Balcisoy and D. Thalmann. Interaction between real and virtual humans in augmented reality. In *CA 1997*, page 31, 1997. 2
- [2] B. G. Baumgart. *Geometric modeling for computer vision*. PhD thesis, Stanford University, Stanford, CA, USA, 1974. 1
- [3] J.-Y. Bouguet. Camera calibration toolbox for matlab, 2006. [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/). 5
- [4] K. Cheung, T. Kanade, J. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *CVPR 2000*, pages 714 – 720, 2000. 2
- [5] S. S. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. *Proceedings of Video Communications and Image Processing*, pages 881–892, 2004. 2
- [6] T. Cuypers, Y. Francken, and P. Bekaert. Shadow multiplexing for single camera visual hull reconstruction. In *PRO-CAMS '08: Proceedings of the 5th ACM/IEEE International Workshop on Projector camera systems*, pages 1–2, 2008. 1, 2
- [7] T. Cuypers, C. Vanacken, Y. Francken, F. V. Reeth, and P. Bekaert. A multi-camera framework for interactive videogames. In *GRAPP 2008*, pages 443–449, 2008. 2
- [8] B. D. Decker, T. Mertens, and P. Bekaert. Interactive collision detection for free-viewpoint video. In *GRAPP 2007*, pages 114–120, 2007. 2, 6
- [9] Y. Francken, T. Cuypers, T. Mertens, J. Gielis, and P. Bekaert. High quality mesostructure acquisition using specularities. In *CVPR 2008*, pages 1–7, 2008. 2
- [10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 5
- [11] J.-M. Hasenfratz, M. Lapierre, and F. Sillion. A real-time system for full body interaction with virtual worlds. *Eurographics Symposium on Virtual Environments*, pages 147–156, 2004. 2
- [12] M. Hatzitheodorou. Shape from shadows. *Journal of Complexity*, pages 63–84, 1998. 2
- [13] B. Horn. Obtaining shape from shading information. In *The Psychology of Computer Vision*, pages 115–155, 1975. 2
- [14] P.-H. Huang and S.-H. Lai. Contour-based structure from reflection. *CVPR 2006*, pages 379–386, 2006. 2
- [15] H. Kawasaki, R. Furukawa, R. Sagawa, and Y. Yagi. Dynamic scene shape reconstruction using a single structured light pattern. In *CVPR 2008*, pages 1–8, 2008. 2
- [16] D. Lanman, R. Raskar, A. Agrawal, and G. Taubin. Shield fields: Modeling and capturing 3d occluders. *SIGGRAPH Asia 2008*, 27(5), 2008. 1, 2
- [17] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE PAMI 1994*, pages 150–162, 1994. 1
- [18] B. Lok, S. Naik, M. Whitton, and J. F.P. Brooks. Incorporating dynamic real objects into immersive virtual environments. In *I3D 2003*, pages 31–40, 2003. 2
- [19] A. M. McIvor. Background subtraction techniques. *Image and Vision Computing*, 2000. 2
- [20] Osram. Osram opto semiconductors - products, 2008. [http://www.osram-os.com/osram\\_os/EN/Products](http://www.osram-os.com/osram_os/EN/Products). 4
- [21] PointGreyResearch. Imaging products - grasshopper, 2008. <http://www.ptgrey.com/products/grasshopper>. 4
- [22] S. Savarese, H. Rushmeier, F. Bernardini, and P. Perona. Shadow carving. *ICCV 2001*, pages 190–197, 2001. 1, 2
- [23] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. *CVPR 2003*, pages 195–202, 2003. 2, 5
- [24] Y. Schechner, S. Nayar, and P. Belhumeur. A theory of multiplexed illumination. *ICCV 2003*, pages 808–815, 2003. 3
- [25] A. R. Smith and J. F. Blinn. Blue screen matting. In *SIGGRAPH 1996*, pages 259–268, 1996. 2
- [26] R. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, pages 139–144, 1980. 2
- [27] S. Yamazaki, S. G. Narasimhan, S. Baker, and T. Kanade. Coplanar shadowgrams for acquiring visual hulls of intricate objects. In *ICCV 2007*, pages 1–8, 2007. 1, 2, 5

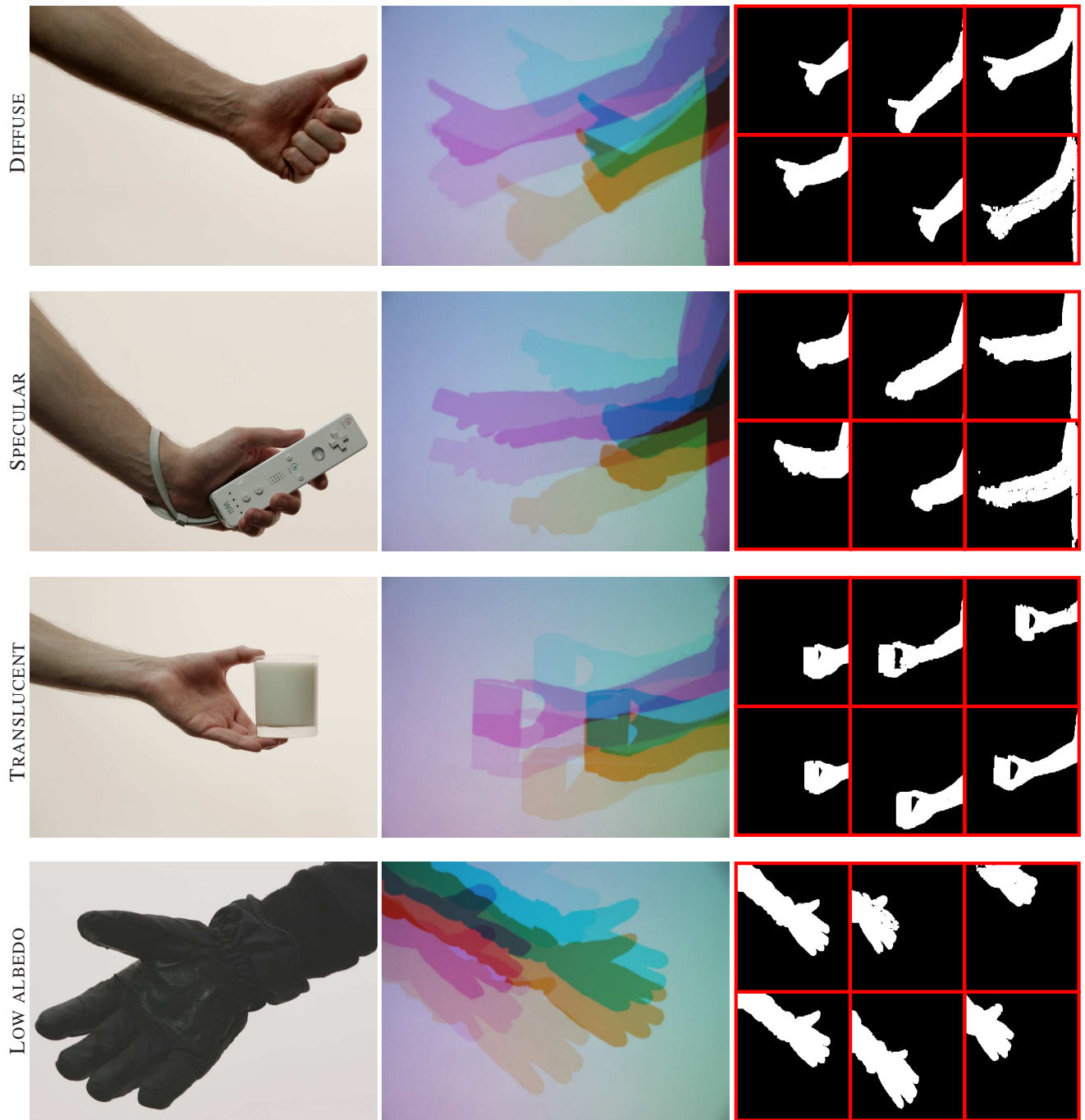


Figure 7. Left: The object we want to obtain multi-viewpoint silhouettes from, middle: Input image consisting of multiplexed shadows, right: Six demultiplexed shadows