Image segmentation assisted by a triple-camera setup

Na Liu

Delft University of Technology
The Netherlands
Image segmentation assisted by a triple-camera setup

by

Na Liu

to obtain the degree of Master of Science at the Delft University of Technology, to be defended publicly on February 27, 2018.

Student number: 4615689
Project duration: March 1, 2017 – February 27, 2018
Thesis committee: Prof. Dr. Elmar Eisemann, TU Delft, supervisor
Dr. Anna Vilanova, TU Delft
Dr. ir. Hans Tonino, TU Delft

This thesis is confidential and cannot be made public until February 22, 2018.

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Dr. Elmar Eisemann from the Computer Graphics and Visualization group at the faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) of the Delft University of Technology for his continuous support, bright ideas and immense knowledge. His guidance helped me to build up and finish this master thesis project.

I would also like to express high appreciation to my coordinator Dr. Anna Vilanova for her timely help and positive encouragement through the whole second year of my master study.

I am very grateful to my daily supervisor Liao Jingtang for his patient explanation and encouragement during my work of this project. Thanks also go to all my friends. It is their care and help that kept me moving forward.

Most importantly, none of this could have happened without the generous support and unconditional love from my beloved parents. I would like to thank and dedicate my master thesis to them, who always stand besides me throughout my whole life.
Abstract

Photographers suffer from many challenging problems when separating and compositing images, such as the color camouflage and unpleasant color bleeding artifacts. To tackle these issues, we present a solution to fore- and background segmentation for images taken from multiple cameras, which introduce the parallax that leads to the foreground model projecting to different background locations, which has been designed to exhibit a special pattern to ease the object extraction. We optimize the camera distance and model position to ensure the silhouette of the model falls on different colors of the background pattern on the wall. Furthermore, we develop a visualization tool based on our algorithm to help photographers effectively setup their studios.
## Contents

1 Introduction ........................................... 1

2 Related Work ........................................ 5
   2.1 Color camouflage .................................... 6
   2.2 Semitransparent occlusion ............................ 8
   2.3 Complicated foregrounds and backgrounds ............ 9
   2.4 Shadow effects ...................................... 10

3 Problem Formulation .................................... 13

4 System Setup .......................................... 19

5 Our Approach .......................................... 25
   5.1 Method overview .................................... 25
   5.2 Transformation pipeline ............................... 27
      5.2.1 Perspective projection ............................ 29
      5.2.2 Viewport transformation .......................... 31
   5.3 Parallax calculation ................................ 31
   5.4 Silhouette error .................................... 33
   5.5 Resolution of the object ............................. 34
   5.6 Background pattern design ........................... 35
   5.7 Cameras distance and model location optimization ... 36

6 Segmentation Model .................................... 37
   6.1 Background subtraction ............................... 37
   6.2 Foreground alignment ................................ 39

7 Implementation ....................................... 41
   7.1 System design ....................................... 41
      7.1.1 Integrated Development Environment ............. 41
      7.1.2 Qt framework .................................... 41
      7.1.3 Open Graphics Library ............................ 43
   7.2 System features ..................................... 43
      7.2.1 Multiple views of the same model ............... 43
      7.2.2 Simultaneous views ............................... 43
7.2.3 Customization .................................................. 43
7.2.4 Direct manipulation .......................................... 44
7.2.5 Two dimensional canvas ................................... 44

8 Results ................................................................ 45

9 Limitations and Discussion ........................................ 53
  9.1 Illumination conditions ....................................... 53
  9.2 Multiple images .................................................. 54
  9.3 Multiple cameras ............................................... 54

10 Conclusion and Future Work ..................................... 55

Bibliography .............................................................. 57

A Formulas Derivation of Silhouette Error and Resolution ... 65

B Parallax Formula Derivation ...................................... 67
Introduction

Image composition has received enormous attention in the past years. Specifically, in the context of a photographic studio, image composition is of great interest to photographers, which is a common task of putting together a foreground object (mostly a person) from images taken in a shooting studio and a new background. The resulting composed images are useful in many contexts, such as advertisement or the movie industry. Image composition includes adding things to shot, removing the elements from the shots, usually the background, and then blending the assets to make them look like consistent. Key is the fore- and background separation. Image segmentation plays an essential role in the field of computer vision. It refers to the process of meaningfully assigning every pixel a label (fore- or background) and then clustering the pixels with same label together, which has been widely used in various applications, such as, medical imaging analysis [44][42], photograph composition in the shooting studio [45], object detection [40][25], etc.

A conventional solution to fore- and background separation for the studio imaging relies on chroma key, which is analogous to terms such as color keying or color separation overlay. In theory, any color that is uniform and distinct can be applied for a chroma key background whereas the colors of green and blue are the most common choices in practice, which is due to the fact that blue and green differ most distinctly in hue from human skin colors [3], and they are both primary colors and can be easily isolated and electronically subtracted from the images [19]. Unfortunately, the solution is not always fail-safe since foreground objects may share similar color with the background, which is namely the color camouflage problem [31]. It is
one of the challenging issues in the field of fore- and background segmentation, and leads to holes in the computed foreground. In addition, both blue and green screens have what is known as color spill or bleeding which can change the image color balance. The color spill is the color that reflects onto and bleeds into the foreground object. Especially, the chromakey green can produce the undesirable "green screen glow" or fuzzy edges in some captured scenarios [56].

There is a large body of literature on the color camouflage, which is typically divided into two main categories, i.e., special-designed background based methods and multiple cameras based methods. On the one hand, a background wall with a special-designed pattern, such as two-tone striped area [66] and checker pattern [3] other than a uniform color, is a good choice to solve the camouflage problem. However, these methods will fail when the model wear the clothes that have the same pattern as the background, additionally, these methods typically assume that the foreground object does not exhibit any holes. The latter is due to the methods trying to guess outline points inbetween, which can be very inaccurate. On the other hand, the depth information is available from depth cameras, such as stereo cameras [31][41][13], which can be used to resolve the ambiguities of RGB descriptors. However, these techniques have limitations due to the accuracy of depth sensors. So they are not widely used in the real life applications. Moreover, depth camouflage errors occur when foreground and background share similar depth information.

In this thesis, we propose a novel method to fore-and background separation for images taken from a shooting studio based on the combination of a background wall with a special pattern (black and white striped areas) and multiple cameras. The pattern will ensure the separation. Nonetheless, as it is mentioned above, a simple pattern is insufficient, as it could still happen that the foreground object shares colors with the background. To address this issue, we will install several cameras at different positions. The idea is that these separate locations will introduce parallax [15] that will lead to the model projecting to a different background location. In consequence, by having chosen the right distance between the cameras and model position in accordance with the background pattern, there will be several input photos available for which the silhouette of the model will fall on different colors of the background pattern on the wall, which helps in the fore- and background separation process. Hereby, the camouflage problem can be solved. In addition, we choose black and white pattern, so there is no hue
influence.

Nevertheless, there are several challenges in this setup. First, changing
the camera location means that the silhouette of the object changes. We aim
at minimizing this effect, while ensuring that the projection will still respect
the background pattern. Second, the minimal number of cameras needs to
be determined. At the same time, we want to include other constraints, such
as the resolution of the object to optimize image quality.

To this end, we propose an optimization procedure that integrates multi-
ple camera setups and background patterns. We also present some insights
on how to use the images taken from multiple cameras to assist the fore-
and background segmentation.

Overall, we make the following contributions:

• We propose a novel method to assist image segmentation and try to
avoid the color camouflage and color bleeding problems. We show that
with the help of three cameras and special-designed background pat-
tern, it is possible to get better segmentation performance.

• We develop a framework to help the photographers effectively setup
their shooting studio by providing essential information about what
pattern the background wall should be painted in and where the model
and the cameras should be placed, once knowing the size of their stu-
dios.
Related Work

In the field of photographic studio imaging, the photographers use different techniques for editing photographs. One common example are compositing techniques using chroma key, which plays an indispensable role in photography processing and video production. In the chroma key compositing technique, the foreground object is photographed by cameras in front of a mono chrome backdrop. Then the background of the captured image is separated from the foreground using image segmentation algorithms, and thereafter the background is made transparent or replaced by any desired background image to create the final image. Generally, the chrome backdrop typically includes single color, such as green, blue, magenta, etc. One prevailing application of chroma key compositing technique is television broadcasting, where the weather reporter stands against the chrome backdrop and then chroma key compositing technique is used to replace the chrome backdrop with a weather map [36].

In the process of image compositing, one essential part is to separate the target elements from the captured image. There are many techniques developed for image segmentation. However, achieving a good or perfect segmentation result is still challenging. There are four common challenges, specifically in the context of photographic studios as shown in Fig. 2.1. First is the color camouflage. The second one is the semitransparent occlusion. Third, it is difficult to handle the complex foregrounds and backgrounds. Last are shadows. Different criteria can be applied when segmentation algorithms are classified, such as excluding or including depth data, unsupervised or supervised methods, single images or multiple images input. In this thesis, the existing image segmentation methods are classified into four
2. Related Work

main categories based on what major problems they aim to solve.

![Figure 2.1](image-url) Common challenging scenarios in shooting studios. (a) Color camouflage. If the model wears a cape of green standing in front of a green screen, there will be only a floating head left in the segmentation result, which is cool for the special effects in the movie but not for the typical on-camera presentation. (b) Foreground with transparent parts [60]. (c) Foreground images with fuzzy boundary [67]. (d) Shadow effect. If the model is too close to the green screen or the light from the front is too strong, a dark shadow might be cast onto the green screen.

2.1. Color camouflage

In an ideal chroma key setting, there should be a significant difference between the color of fore- and background. However, in real-world productions, the range of foreground colors may overlap the range of background colors [28]. We introduce two basic types of segmentation techniques that aim to solve the color camouflage in this section: one is an interactive method, the other is an RGBD approach without any user interaction.

Some studies [10][32] about interactive image segmentation concentrate on capturing the tendency for images of solid objects to be coherent via using both color and contrast information as well as Markov Random Field priors. In these works, transparent objects are not allowed. The foreground estimate is obtained precisely by a graph cut algorithm. It even handles cases where the background and foreground color distributions partly overlap.

One novel paper [7] addresses the camouflage problem with minimal user interactions. Again, the transparency is not allowed. The user needs to draw a fat pen trail enclosing the object boundary, which defines the train-
2.1. Color camouflage

ing "trimap" with the foreground or background or unclassified labels. All the pixels in trimap region need to be classified fore- and background. The graph cut algorithm is applied to the unclassified region by using the learned parameters from the models for color and texture properties of the fore- and background pixels. Finally, good object segmentation results are generated.

RGBD images segmentation techniques without interaction also aim to reduce the ambiguities of RGB descriptors by taking advantage of depth information. The overall idea in RGBD segmentation is building a joint model to combine color and depth information and to choose an optimal classic segmentation technique. The depth data can be provided by stereo cameras, Kinect camera, Time-of-Flight camera and two or three types combination. There are already a lot of traditional image segmentation algorithms that had been previously developed for color or intensity images, which have been enhanced to incorporate depth information [34]. For example, the graph based segmentation [24] is extended in [53] to fuse color with depth. The gPb-UCM algorithm [4] is extended in [47] to the RGBD images. Another work [37] efficiently applied clustering (K-means) method to RGBD pixel values. Further, many recent works on RGBD image segmentation [21][33][51][30] have shown that depth as an additional feature certainly improves the performance of segmentation methods and especially solves the ambiguities of color descriptors.

Although depth data allows for a more robust and reliable segmentation of the objects of interest with respect to the color camouflage problem than a system only based on color, these systems will exhibit other problems; every depth camera has limitations and may lack precision. Stereo vision often fails in textureless regions and for repeated patterns. Cameras like Time-of-Flight and Kinect, can deal with those problems by emitting signals [29], but they are low-resolution, noisy and become unreliable interfered by the ambient light and scene reflectance where passive stereo methods work well. Objects contours will be heavily affected by the high level of depth-data noise at object boundaries [11][12]. Hence, the combination of different depth cameras can help to compensate each other and make the results more robust [61].

One impressive example is a system which addresses high resolution and high quality depth estimation based on joint fusion of stereo and Kinect data [68]. The result shows that the system outperforms either the conventional stereo vision or Kinect alone, which can provide high quality and resolution depth map by complementing the strength of stereo vision and the
Kinect depth sensor. However, the system construction is complicated and it will be difficult and expensive to reproduce the system. Additionally, the stereo cameras need to be calibrated first using the standard checkerboard-based method [69]. This complicated calibration may prevent the practical usage of stereo cameras to provide depth map.

2.2. Semitransparent occlusion

Since an object is still visible when placed behind semi-transparent objects, such as glass, it is hard to define whether the seen pixel belongs to the semi-transparent objects or the object behind. There are many studies [50] [8] that focus on inference of the transparency to deal with mixed pixels and translucent textures.

One famous technique is digital matting, where the foreground objects are extracted from the background image via estimating the color and opacity for the foreground objects at each pixel. The opacity value is called alpha. Fractional opacity values (between 0 and 1) are important for the transparency and motion blur of the foreground objects. Matting is often used to composite the foreground elements into a new scene, which was originally developed for the video and film production [22]. In matting and compositing, the synthetic image $I$ can be composed by the compositing equation with the foreground color $F$, the pixel-wise background color $B$, and the alpha channel $\alpha$ [45]:

$$I = \alpha F + (1 - \alpha)B \quad (2.1)$$

The observed color of every pixel in the image is decomposed into two parts based on the above equation: the foreground color and the background color. However, in real life, the colors of foreground and background are not always separately distributed. Color ambiguity between fore- and background pixels is inevitable at blurred edges, in translucent regions, etc. So the observed color could also be the result of mixing the foreground color and the background color.

In the absolute foreground region, the observed pixel color is a foreground color, and the value of $\alpha$ is 1. In the absolute background region, the observed pixel color is complete background color, and the value of $\alpha$ is equal to 0. In the regions where the fore- and background colors mix, the value of $\alpha$ varies from 0 to 1. The specific value of $\alpha$ for each mixed pixel can be determined by the local color mixture condition. Based on the color mixing...
model, the foreground elements along with their transparency property can be accurately and efficiently estimated and then extracted from the background.

In [60], a chroma keying combining alpha matting system is proposed that can automatically estimate the alpha map and the reliable intrinsic color of foreground objects in front of solid background. It can distinguish the translucent foreground from even reflective foreground and shaded background, which effectively improves the accuracy and reliability of transparency estimation.

Another iterative optimization approach is presented in [59]. The method is based on a small sample of fore- and background pixels marked by the users, the opacity value for every pixel is iteratively estimated. A well specified trimap is not required in this method. Thus it is more efficient to extract the high quality mattes for the foregrounds with semi-transparent regions or the foreground objects with many holes where the trimap is difficult to create even manually.

2.3. Complicated foregrounds and backgrounds
For some complex images, the problem of reliably distinguishing the regions that contain fine details such as fuzzy edges and trivial lines (e.g., hairs) is the most important and challenging. There are various techniques and systems [16][54][23][55][48] proposed to extract high quality complex foreground.

One impressive technique is the Flash Matting [55] proposed by Sun et al. that extracts a high-quality matte by using a flash photo and non-flash photo pair. It is based on the assumption that the appearance of foreground may change drastically under the camera flash, which allows reliable fore- and background separation. Then a joint Bayesian flash matting algorithm is developed and applied to the non-flash image and flash-only image. Moreover, both the trimap and matting results can be quickly and automatically obtained even for highly complex objects. However, it is difficult to extract the high quality mattes when the fine structures in the foreground move and the input image pair can not be pixel-aligned.

There is also a lot of research that incorporates some user intervention to extract the fine structures, such as hair strands or blurry edges. For instance, the papers [6][46] present an interactive framework based on the user-provided scribbles to label the regions of interest. The weighted geodesic distances from every pixel to the scribbles are computed, from which the
images and videos are automatically segmented. The weight is the gradient of the likelihood that a pixel belongs to the foreground (resp. background). This algorithm can accurately handle diverse data with fine structures, but it works better only when the pixel value distributions between fore- and background do not overlap.

Recently, significant accuracy in image segmentation has been achieved by using artificial neural networks [38] trained by back-propagation [49] even when the foreground objects have complicated structures or they are randomly distributed and non-compact in the images. These artificial neural networks are classifiers which are inspired by biologic neurons [57]. Every single artificial neuron in neural networks has some inputs which are weighted and summed up. Then the neurons apply an activation function to the weighted sum, which gives an output. Those neurons can take either a feature vector as input or the output of other neurons. In this way, they build up feature hierarchies.

There are many successful segmentation applications around neural networks. For instance, Long et al. [39] showed that convolutional network architectures, which were originally developed for the image classification, can be successfully redesigned for a dense prediction. Furthermore, these repurposed networks substantially outperform competitors even on challenging semantic segmentation benchmarks. In paper [14], detailed segmentation maps along object boundaries and semantically accurate predictions are achieved by combining the deep convolutional neural networks and fully-connected conditional random fields, which is well beyond the reach of existing methods. Moreover, a more recent paper [20] shows that segmentation with much deeper networks is also possible and can achieve even better results for the complex images.

2.4. Shadow effects
In the context of the shooting studio, the shadow effect is another main challenging problem for the fore- and background segmentation algorithms. Because it changes the appearance of the scene and consequently makes the foreground objects to be regarded as the background. A perfect chroma screen lighting is even with no signs of shadows or creases. The objects need to be far enough away from the screen to prevent shadows and bleeding. Bleeding is when the color of the chroma screen bounces onto the foreground subjects. Typically, a good distance is five feet or further away from the chroma screen to avoid spill and ten or more feet to avoid shadows in
the shooting studios [9].

For the natural images, shadow effect related with sunlight is still a major difficulty, which can not be avoided. There are various studies [58][17][65] that aim to remove the shadows from the natural images. Via shadow removal techniques, unwanted shadows will be detected and then eliminated from the image in order to obtain a shadow-free image. For example, in the work [43], texture-illumination decoupling is used to remove the soft shadows on textured surfaces of an image, followed by bilateral filtering. The paper [62] presents a method with a sequence of $N$ distinct images with differing illumination to derive one reflectance image that is free of cast shadows. In [26], a method is proposed to remove shadows from a single image so as to extract the meaningful information from an image and produce an aesthetic image free of shadows. An automatic shadow removal algorithm [27] is proposed by minimizing entropy to derive a resulting illumination invariant image for shadow removal. Paper [5] proposes a method to remove shadows from real images based on shadow density, where an image is segmented into sunshine, penumbra and umbra regions, followed by color and brightness adjustments for shadow removal. In [65], the approach provides a new interactive technique for shadow markup to create a quadmap, then the authors minimize a new optimization shadow equation that includes a color term and a smoothness term to remove the natural shadows.
Problem Formulation

In this chapter, we formulate our solution to color camouflage and bleeding problems, which often occur during the photographers segmenting and compositing images taken from the shooting studio. We also give a summary of our method and briefly introduce our system.

There are various considerations in setting up a studio to get high-quality photos, which can be used to facilitate the subsequent edition, fore- and background segmentation and composition processes. For instance, a photographer might concern about if the room size is suitable for portrait shooting, and what kind of pattern the background wall should be painted in in order to reduce the ambiguity of color camouflage and avoid color bleeding artifacts. They may think about how many cameras they need, what kind of cameras they should choose and where the cameras should be placed. Perhaps, they might also consider where the model should stand.

Therefore, we design a system to provide the above essential information for setting up a studio. The photos taken in our special-designed configurations can efficiently assist the fore- and background separation process to be free of the color camouflage and bleeding problems. Our ultimate goal is to help the photographers accurately get a top-quality foreground image without background.

Our analysis starts with a collection of simplifying assumptions. The multiple cameras used in our system lie on a horizontal line that is parallel to the background wall. The orientations of the cameras are orthogonal to the wall, which are not allowed to change. The model always stands on the central line of the multiple cameras. We assume the dimension of the captured image in our system is $1024 \times 1024$ pixels at 72 ppi. Because the com-
mon dimension choice of the good quality photographs usually requires at least 1024 * 1024 pixels, and an image resolution of at least 72 pixels per inch (ppi) could provide good results in industry standard for printing a photo or for web use. Our system can be extended to work for the image with higher resolution and dimension.

In addition, we use standard lens for the cameras in this thesis, also known as normal lens, which has a roughly same angle of view as what the human eye can comfortably view. Standard lenses are a common choice for the photographers, because they can give photos a pleasing, natural feel and help to focus attention on the object, instead of distracting the observers with an unusually distorted image [2]. Standard lenses have three main characteristics. First, they can produce a natural perspective because of their similar angle of view to the human eye. Angle of view is the simulated viewing angle through the camera, which describes that part of the world that is visible through the camera at a particular position and orientation in space; objects outside the angle of view when the picture is taken will not be recorded in the photograph. Second, for a full-frame camera with a standard lens, the most commonly used focal length is 50mm (focal length is the axial distance from the camera center to the image plane). Lenses with longer or shorter focal lengths result in an expanded or contracted field of view that appears to distort the perspective when viewed from a normal viewing distance [52][64]. Third, standard lenses are "prime" (i.e., they have a fixed focal length). This may deter some photographers, who think that standard lenses are less versatile than zoom lenses. However, the standard lenses more than make up for the lack of zoom by having superior optical quality and wide apertures. This means the standard lenses can capture stunning images in a wide range of situations and lighting conditions [2].

Another simplifying assumption is that we regard the real human body as being made up of a cylinder with varying radius from head to feet. The distance from every point of the silhouette to the center line of a human model (represented as yellow line in Fig. 3.1) is deemed to be the radius of the model, which changes from top to bottom. We assume that the largest radius $r$ equals 0.3 meters, then the largest diameter is the largest breadth of an adult. So the radius of the model varies from 0 to 0.3m in this thesis. For instance, in Fig. 3.1 (c), point A is one of the leftmost silhouette points of the human arm. We define the distance between A and the center point O of the human model as the radius instead of the distance between A and the center point O' of the arm. Likewise, for the rightmost silhouette point
B of the arm, we assign the distance $BO$ instead of $BO'$ to the radius. The reasons for this assumption are two-fold. On the one hand, the centers of every transversal cross section will always be on the same center line of the human model, only the radius of the transversal cross sections is varying. Otherwise, it will need more than one formula to represent the parallax. On the other hand, this assumption aims to benefit the following mathematic calculations, which are deducted near the silhouette points, such as point A. Because the maximum radius defined in our assumption model is larger than the arm radius, the values computed with the arm center and arm radius will be included within the range calculated by our model. Despite the complicated structure of the human model defined in this thesis, it can ultimately boil down to the simple circle viewed from the top. As for the foreground model with holes, our method can also support. Because we apply the background subtraction technique to separate the foreground object. The background pattern will exhibit through the holes of the foreground object, which will be subtracted during the segmentation process.

![Figure 3.1: Human body can be modeled by a cylinder with changing radius from the top to bottom.](image)

Based on the above assumptions, we designed our system as shown in Fig. 3.2. Firstly, our system helps the photographers determine the minimal number of cameras that is the same for all the studios. Due to these
different camera locations, the **parallax** is introduced, which we define as the difference or displacement in the apparent position of the foreground object on the background wall viewed from the multiple cameras. The parallax will enable the foreground object to project on different locations of the background wall.

**Figure 3.2:** Overview: the goal of our system is to help the photographers effectively setup their shooting studio and get high-quality extracted foreground without the trouble of color camouflage. Given a studio of size $H \times W$, we design a special striped background pattern with specific frequency. We will also give guidance about where the cameras and model should stand for shooting images, which aims at helping the photographers get high-quality photos and precisely extract the foreground model.

Secondly, we seek to design a special pattern for the background wall based on the studio size, which is painted only once. The **input** of our system is the dimension of a given studio $H \times W$, The studio dimension is not necessarily the exact studio room size in this thesis. It means the space that the photographers plan to use for taking photos. The photographers can reserve some other spaces in their studio room for placing professional photographic equipment and so on. We will also show if the input studio suits our solution. One of the **outputs** of our algorithm is the background pattern. This special-designed pattern, combined with the parallax of the multiple camera setup, will support the image segmentation by ensuring that the target object falls on totally different colors of the background pattern on the wall, hereby solving the color camouflage problem. A single color is insufficient, but we show that a simplest two-tone pattern with black and white stripes suffices. The black and white colors can also avoid the unpleasant color bleeding artifacts. We choose a vertical striped pattern with a specific frequency, as we assume that cameras will be aligned horizontally, which means that their parallax shifts the object horizontally. The frequency is related to the specific dimension of the studio. In addition, which color should be painted first does not matter, as long as the pattern frequency is fulfilled.

At last, other outputs of our system are the camera distance and model position in order to help the photographers effectively setup their studio and get high-quality images, which are used for fore- and background sep-
In the following chapters, the setup of our system is introduced (Chap. 4), describing how we derived the solution and determining the minimal number of cameras we need to use and assist the image segmentation. We then demonstrate our algorithm (Chap. 5). The core of our method builds upon a parallax analysis. To this extent, we build formulations for calculating the parallax (Sec. 5.3), the silhouette error (Sec. 5.4) and the resolution of the object (Sec. 5.5). Next we discuss the background wall pattern design process (Sec. 5.6) and optimize the camera distance and model position (Sec. 5.7) on the basis of the parallax analysis. Then we present some insights on how to use the images taken from multiple cameras to assist the fore- and background separation (Chap. 6). We also explain how we implemented our system (Chap. 7) and show the results (Chap. 8). We finally discuss the limitations of our algorithm (Chap. 9), before concluding (Chap. 10).
In this chapter, we describe the setup of our system, illustrating how our idea is generated step by step.

We initially chose **stereo cameras** to assist image segmentation process and solve the color camouflage problem. We take advantage of the parallax introduced by the multiple cameras, which is caused by actual change (or difference) of camera position that provides a new view direction. In this thesis, the **parallax** is measured by the distance or displacement of the projection points of the object silhouette on the background wall via the multiple different cameras. For instance, in Fig. 4.1, $A_1$ and $A_2$ are the leftmost silhouette points in the view of Camera 1 and Camera 2, respectively. $P_{A_1}$ and $P_{A_2}$ are the projection points of the corresponding silhouette points on the wall via the stereo cameras. Parallax is measured by the distance between the projection points $P_{A_1}$ and $P_{A_2}$ on the background wall.

**Figure 4.1:** Stereo Case: our system initially consists of stereo cameras, a background wall with a special-designed pattern, and a model in-between. **Left:** side view of our system. **Middle:** top view of our system. **Right:** two images taken from stereo cameras; the first image is the view of Camera 1, and the second one is the view of Camera 2. The model appears to be in front of the white background wall from the view of Camera 1, but appears to have moved in front of the black wall in the view of Camera 2. This displacement or difference in the apparent position of an object on the background wall via different cameras is called parallax.

Based on the principle of parallax, we wish to ensure that for the two taken photos, every point on the silhouette of the target object projects on a
different color of the background pattern to avoid color camouflage. At first, we applied the most simple two-tone pattern to paint the background wall as illustrated in Fig. 4.1. Theoretically, any two kinds of colors with great contrast can be used. However, black and white can more effectively avoid hue bleeding, so we paint half of the background wall with black color, and the rest remains white.

On the basis of parallax analysis and a special background pattern, the ideal situation is that we obtain one image with black background, and the other with white background by choosing the right camera distance and model position. These two images should be the same except the background color. Both of them are supposed to help us solve the color camouflage in the segmentation process. For instance, in the Fig. 4.1, the image taken from Camera 1 can better help us separate the black pants from white background, while the image taken from Camera 2 is more helpful in segmenting the white shirt from black background. The final segmentation result is the combination of the above two separation results that are the complete black pants and complete white shirt.

However, the above solution cannot always succeed to provide assistance, because some parts of the object may fall on the black background and meanwhile the other parts project on the white background in both two views of the cameras. Besides, we wish to keep the camera distance small; otherwise, the silhouette of the object will change during the segmentation process. So we decrease the distance between stereo cameras. With the cameras moving toward each other, the parallax goes down. The background pattern should also change to work together with the parallax. For simpleness, we choose the two-tone striped pattern whose stripes are in the same size as shown in Fig. 4.2. The vertical stripes are used since all the cameras are aligned horizontally, which causes the parallax needs to shift in the horizontal direction, so does the background pattern.

![Figure 4.2: Stereo Case-2: when the distance between two cameras decreases, the parallax becomes smaller. The background pattern should also change to get several images with the foreground model standing in front of different background pattern.](image)

Nevertheless, the stereo cameras, together with the striped pattern, are
still not always fail-safe to provide help. After inputting the studio size, the pattern frequency is designed and fixed at first. We use parameter \( d \) to represent the pattern frequency, which is also the width of every stripe of the background pattern. The parallax value will vary due to the different camera distance, model position and radius. For instance, when the parallax value belongs to the interval \([0, d]\) as shown in Fig. 4.3, the projection points \( P_{A_1} \) and \( P_{A_2} \) of the silhouette might both fall inside one stripe on the background wall via stereo cameras, while another pair of projection points \( P_{B_1} \) and \( P_{B_2} \) may project on two adjacent stripes separately. The former situation cannot assist the segmentation process, while the latter can effectively help to generate the images for which the silhouette of the model will fall on different background colors. Likewise, when the parallax value varies in the following intervals of one \( d \), there is always a possibility that these projection points pair fall on two stripes with the same color. This possible situation may happen and fail our method for all the parallax value, and no constraint can be included to avoid it.

\[
\text{Parallax } \in [0,d] \quad \text{Parallax } \in [d,2d] \quad \text{Parallax } \in [2d,3d]
\]

Figure 4.3: Failure cases by using stereo cameras: Left: Parallax \( \in [0, d] \). Middle: Parallax \( \in [d, 2d] \). Right: Parallax \( \in [2d, 3d] \). Parameter \( d \) is the pattern frequency. There is no parallax value that can always absolutely ensure the projection points of the silhouette fall on the stripes with different colors of the background pattern via the stereo cameras.

Therefore, we introduce a third camera in the middle to prevent the above possible circumstances and make compensation as illustrated in Fig. 4.4. Placing the third camera in the middle position is easy to manipulate and implement. Since we have three cameras, two more parallaxes are generated. The parallax caused by Camera 1 and 2 is the biggest one, which we name as \( P_{12} \). The others are named \( P_{13} \) and \( P_{32} \). If the target object is a plane, the parallax value \( P_{13} \) equals \( P_{32} \) according to the triangle similarity theorem. The projection point \( P_{A_3} \) will precisely fall in the middle between \( P_{A_1} \) and \( P_{A_2} \). For the object with a curved surface, such as the human model, these two parallax values are not always same. They are equal to each other when the model radius is zero. Otherwise, one of them must be larger than the other.

Since the parallax \( P_{12} \) varies due to the different camera distance, model position and radius, we need to find what conditions \( P_{12} \) should satisfy in
order to always assist the segmentation process without failure. We piecewise explore the relationship between $P_{12}$ and the pattern frequency $d$ in the following:

- For the parallax value $P_{12} \in [0, d]$, two projection possibilities show in Fig. 4.5. If projection points $P_{A_1}$ and $P_{A_2}$ fall on the stripes with different colors of the background pattern, the color camouflage problem can be solved. However, there is another situation that they are all located inside one stripe.

- For the parallax value $P_{12} \in (d, 2d]$, there are also two possible scenarios as shown in Fig. 4.6. Three projection points may fall on two stripes with the same color (Fig. 4.6 (b)). So these parallax values cannot be always fail-safe to provide help.

- For the parallax value $P_{12} \in (2d, 3d]$, Fig. 4.7 (a) shows that the projection points $P_{A_1}$ and $P_{A_2}$ on the background fall on two stripes with different colors, the color camouflage problem will be resolved no matter what color of stripe the in-between point $P_{A_3}$ falls on. If the projection points $P_{A_1}$ and $P_{A_2}$ fall on two stripes with the same color in Fig. 4.7 (b), the location of point $P_{A_3}$ matters. If the smaller parallax of $P_{13}$ and $P_{32}$ could be larger than pattern frequency $d$, the projection

![Figure 4.4: Triple Cameras Case: Left: side view of our system. Middle: top view of the system. Right: images taken from triple cameras; the first image is the view of Camera 1, the second image is the view of Camera 3, and the third image is the view of Camera 2. When the projection points of the silhouette via Camera 1 and 2 are both projected on the white stripes, the view of the third camera is supposed to assist segmentation by projecting on the black stripe.](image-url)

![Figure 4.5: For the parallax value $P_{12} \in [0, d]$, (a) successful case. (b) failure case.](image-url)

![Figure 4.6: For the parallax value $P_{12} \in (d, 2d]$, three projection points may fall on two stripes with the same color.](image-url)

![Figure 4.7: For the parallax value $P_{12} \in (2d, 3d]$, the projection points $P_{A_1}$ and $P_{A_2}$ on the background fall on two stripes with different colors, the color camouflage problem will be resolved.](image-url)
point $P_{A3}$ will not share the same stripe with $P_{A1}$ or $P_{A2}$, So the three projection points $P_{A1}$, $P_{A2}$ and $P_{A3}$ will fall on three adjacent stripes, among which the middle stripe must have the opposite color from the other two stripes. Under this condition, the color camouflage problem can be solved without failure.

• For the parallax value $P_{12}$ larger than $3 \times d$, likewise, these parallax values still cannot always succeed to ensure that every point of the silhouette projects onto different colors of the background pattern. For instance, Fig. 4.8 shows a failure example that the projection points on the background pattern all fall on the stripes with the same color via the three cameras. Therefore, none of the images taken from the triple cameras can help in the fore- and background segmentation.

Consequently, if we can always keep all the three projection points $P_{A1}$, $P_{A2}$ and $P_{A3}$ falling on three adjacent stripes by adjusting the values of camera distance and model position, we will always get several photos available to assist segmentation, where the silhouette of the model falls on different colors of the background pattern on the wall. So the biggest parallax $P_{12}$ needs to be within $(2d, 3d]$, while the smaller parallax of $P_{13}$ and $P_{32}$ needs to be larger than the pattern frequency $d$. This allows us to formulate the frequency designing as a constrained minimization problem. We choose

![Figure 4.6](image1.png)  ![Figure 4.7](image2.png)
the minimum parallax $P_{13}$ or $P_{32}$ as the background pattern frequency $d$, which should be a specific positive number under some constraints. The parallax value $P_{12}$ is supposed to stay in the range $(2*d, 3*d]$, which guarantees an absolute success to provide effective assistance and also helps us limit the range of camera distance and model location.
5

Our Approach

In this chapter, we describe the algorithm for designing a special background pattern and optimizing the cameras distance and model location, which are based on the parallax analysis.

5.1. Method overview

In order to be easily understood, we setup a simplified system as shown in Figure 5.1, which is the top view of our system. We have three cameras \( C_1 \), \( C_2 \) and \( C_3 \) at the same horizontal plane. We assume that \( c \) represents the camera distance between \( C_1 \) and \( C_2 \). Camera \( C_3 \) locates in the middle of cameras \( C_1 \) and \( C_2 \), aligned with the object center. Parameter \( b \) represents the object position. It is measured by the distance between \( C_3 \) and the center of our target object - the cylinder, which is a circle in 2D, whose center is \( O \) and radius is \( r \) (\( r \) varies from 0 to the maximum value - 0.3m). The leftmost silhouette points of the object are named \( A_1 \), \( A_2 \) and \( A_3 \) in the three views of the cameras \( C_1 \), \( C_2 \) and \( C_3 \). The projection points of \( A_1 \), \( A_2 \) and \( A_3 \) on the background wall are named \( P_{A_1} \), \( P_{A_2} \) and \( P_{A_3} \). We assign the camera \( C_1 \) to the origin of the world coordinates. So the coordinates of \( C_1 \), \( C_2 \), \( C_3 \) and \( O \) are \((0,0,0)\), \((c/2,0,0)\), \((c,0,0)\) and \((c/2,0,b)\).

First of all, given a shooting space \( H \times W \) in a studio, we need spare some distance for model standing away from the background, which helps for flagging off the shadow from getting to the background. Typically, a good distance is about ten or more feet to avoid shadows in the relatively large shooting studios [9]. The same principle applies for smaller space as well. Some professionals suggest that the portrait subjects should be pulled at
5. Our Approach

Figure 5.1: Top view of the system with a background wall, three cameras $C_1$, $C_2$, and $C_3$, and one cylinder $O$ in-between. Blue line $C_1X_1$ is the ray from camera $C_1$, which is tangent to the circle $O$. The intersection point is named $A_1$. The red and the grey tangent lines can be got in the same manner.

### Math Annotations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>Origin - camera on the left, whose world coordinate is (0, 0, 0)</td>
</tr>
<tr>
<td>$C_2$</td>
<td>camera on the right, whose world coordinate is $(c, 0, 0)$</td>
</tr>
<tr>
<td>$C_3$</td>
<td>camera in the middle, whose world coordinate is $(c/2, 0, 0)$</td>
</tr>
<tr>
<td>$c$</td>
<td>distance between the camera $C_1$ and $C_2$</td>
</tr>
<tr>
<td>$O$</td>
<td>our target object - cylinder</td>
</tr>
<tr>
<td>$r$</td>
<td>radius of cylinder</td>
</tr>
<tr>
<td>$b$</td>
<td>distance between the camera $C_3$ and the center of the cylinder</td>
</tr>
<tr>
<td>$H$</td>
<td>length of the shooting studio</td>
</tr>
<tr>
<td>$W$</td>
<td>width of the shooting studio</td>
</tr>
<tr>
<td>$A_1$</td>
<td>leftmost silhouette point in the view of Camera $C_1$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>leftmost silhouette point in the view of Camera $C_2$</td>
</tr>
<tr>
<td>$A_3$</td>
<td>leftmost silhouette point in the view of Camera $C_3$</td>
</tr>
<tr>
<td>$P_{A1}$</td>
<td>projection point of $A_1$ via camera $C_1$ onto the background</td>
</tr>
<tr>
<td>$P_{A2}$</td>
<td>projection point of $A_2$ via camera $C_2$ onto the background</td>
</tr>
<tr>
<td>$P_{A3}$</td>
<td>projection point of $A_3$ via camera $C_3$ onto the background</td>
</tr>
<tr>
<td>$P_{12}$</td>
<td>parallax caused by the cameras $C_1$ and $C_2$</td>
</tr>
<tr>
<td>$P_{13}$</td>
<td>parallax caused by the cameras $C_1$ and $C_3$</td>
</tr>
<tr>
<td>$P_{32}$</td>
<td>parallax caused by the cameras $C_3$ and $C_2$</td>
</tr>
<tr>
<td>$d$</td>
<td>background pattern frequency</td>
</tr>
</tbody>
</table>

Table 5.1: Math Annotations

least 3 feet away from your background to prevent shadows and allow for easy lighting [35] [63]. Therefore, our model moves at least 3 feet (about 0.9 meters) away from the background to keep the shadow from casting on the
background. Besides, model position $b$ is supposed to be larger than the human radius $r$. Otherwise, some parts of the model may disappear in the views of the cameras. So the studio length $H$ essentially needs to be no less than 1.2 meters in this thesis.

Second, although small camera distance will reduce the disparities of the silhouette point in the different views of the multiple cameras, the camera distance cannot be as small as possible. Assuming the resolution of the captured image is $1024 \times 1024$ pixels at 72 ppi in this thesis, there are 72 pixels per inch. So one pixel takes up about 0.035 centimeters. In order to avoid the dense stripes causing dizziness, we want the background pattern frequency to be at least 30 pixels (about 1 centimeter), which will make the camera distance have a minimal value.

We show the overview of our approach with a flow chart in Fig. 5.2. We get the range of camera distances and model positions based on the above two restrictions and the first constraint, which is to achieve the pixel accuracy without sacrificing the resolution of the object. We then formulate the parallax and design a special striped background pattern with specific frequency $d$. Under another constraint that the parallax $P_{12}$ belongs to the range $[2 \times d, 3 \times d]$, we can further optimize the camera distance and model location. If the studio size is smaller than the minimal optimized values, our system cannot be applied. Otherwise, the outputs of our system are the essential appropriate information that the photographers need to setup their studio and get a high-quality image. The images taken in our designed configuration can assist the segmentation process and avoid the color camouflage and bleeding problems.

5.2. Transformation pipeline

We describe the sequence of transformation how 3D world points get projected into 2D pixel coordinates in this thesis as shown in Fig. 5.3. Each step in this sequence can actually be represented by a matrix operation, so the whole process can be applied by matrix multiplication [18]. We call this sequence a transformation pipeline that help us efficiently draw 3D primitives in perspective. $(X, Y, Z)$ are the world coordinates of a 3D point in the world coordinate space. $(x, y)$ are the clip space coordinates of the projection point in image plane. $(u, v)$ are the screen (pixel) coordinates of the projection point. We divide the transformation pipeline into two steps: perspective projection from $(X, Y, Z)$ to $(x, y)$ and viewport transformation from $(x, y)$ to $(u, v)$. Fig. 5.4 intuitively shows our entire transformation pro-
Our Approach

Input: Studio Dimension $H \times W$

$H \geq 1.2$ meters && $W \geq$ Minimal Camera Distance

First Constraint: Pixel Accuracy && Resolution Restriction

Intermediate Output: Wide range of Camera distance and Model Position

Parallax Calculation

Final Output 1: Background pattern frequency $d$

Second Constraint: Parallax value $\in [2 \times d, 3 \times d]$

Final Optimization Output 2: Camera distance and Model Position

$H \geq$ Optimized Min Model Position && $W \geq$ Optimized Min Camera Distance

System Fails

System Succeeds and Stops

Figure 5.2: Flow chart of our approach.
cess where the world origin locates at the camera position and the camera axes is aligned with world coordinate axes.

**Figure 5.3:** Transformation Pipeline [18]. The transformation pipeline can be split into two steps: (a) 3D-to-2D perspective projection and (b) viewport transformation: digitization the image into pixel coordinates.

**Figure 5.4:** Transformation process from the 3D world to 2D pixel coordinates [1].

### 5.2.1. Perspective projection

Given 3D world point \((X, Y, Z)\), we can derive the 2D point \((x, y)\) on the image plane by the following perspective projection equations via similar triangles rule:

\[
x = f \frac{X}{Z} \quad y = f \frac{Y}{Z},
\]

where \(f\) is focal length of the camera. Both \(x\) and \(y\) are defined in clip space (also named Normal Device Coordinates space). That is, the \(x\) and \(y\) should be between -1 and 1; coordinates outside this range will be invisible. These clip coordinates are then given to the rasterizer to transform them to 2D pixels on the screen. In the homogeneous coordinates \((x, y)\) is equal to \((kx, ky, k)\) for any nonzero \(k\). Therefore, the above equations can be represented as a matrix.
\[
\begin{bmatrix}
  x' \\
y' \\
z'
\end{bmatrix}
= 
\begin{bmatrix}
  f & 0 & 0 & 0 \\
  0 & f & 0 & 0 \\
  0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
  X \\
  Y \\
  Z \\
  1
\end{bmatrix}
\]

\( (x = \frac{x'}{z} = f \frac{X}{Z}, \quad y = \frac{y'}{z} = f \frac{Y}{Z}) \) (5.2)

Fig. 5.5 shows a simple two-camera system example. Left camera is assigned to the origin (0, 0, 0) of the world coordinates. The camera axes aligned with world coordinate axes. Right camera locates at world location \((c, 0, 0)\).

In the left camera system, the image coordinates \( (x_l, y_l) \) of the world point \((X, Y, Z)\) can be represented by the following equations:

\[
\begin{bmatrix}
  x_l' \\
y_l' \\
z_l'
\end{bmatrix}
= 
\begin{bmatrix}
  f & 0 & 0 & 0 \\
  0 & f & 0 & 0 \\
  0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
  X \\
  Y \\
  Z \\
  1
\end{bmatrix}
\]

\( (x_l = \frac{x_l'}{z_l} = f \frac{X}{Z}, \quad y_l = \frac{y_l'}{z_l} = f \frac{Y}{Z}) \) (5.3)

In the right camera view, we need firstly transform from the world coordinates to the right camera coordinates system, which can be achieved by a translation matrix. The image coordinates \( (x_r, y_r) \) of the same world point can be represented by the following equations:
5.3. Parallax calculation

\[
\begin{bmatrix}
  x'_r \\
  y'_r \\
  z'_r
\end{bmatrix} =
\begin{bmatrix}
  f & 0 & 0 & 0 \\
  0 & f & 0 & 0 \\
  0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
  1 & 0 & 0 & -c \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  X \\
  Y \\
  Z \\
  1
\end{bmatrix}
\]

\[
(x_r = \frac{x'_r}{z'_r} = f \frac{X - c}{Z} \quad y_r = \frac{y'_r}{z'_r} = f \frac{Y}{Z})
\]

(5.4)

5.2.2. Viewport transformation
The viewport transformation maps the clip coordinates in the normalized space (from -1 to 1) to the screen (pixel) coordinates by the appropriate scale and shift. The image dimension used in this thesis is 1024 \(\times\) 1024 pixels. By doing a linear mapping from \([-1, -1] \times [1, 1]\) to \([0, 0] \times [1023, 1023]\), we can get the screen coordinates \((u, v)\) in pixels:

\[
u = \frac{x - (-1)}{2} \times 1023 \quad v = \frac{y - (-1)}{2} \times 1023
\]

(5.5)

5.3. Parallax calculation

To design the background pattern and find proper cameras and model location, we need to first analyze and formulate the parallax. We define parallax as the difference or displacement in the apparent position of the foreground object on the background wall viewed from the multiple cameras in this thesis. For instance, Fig. 5.6 shows the parallax \(P_{12}\) caused by cameras \(C_1\) and \(C_2\) is mathematically measured by the distance between the projec-
Our Approach

Projection points $P_{A_1}$ and $P_{A_2}$ on the background wall, which is calculated in the 3D world coordinates. According to the triangular geometry, we can calculate the parallax by the following formula:

$$P_{12} = \frac{H \cdot c \cdot b}{b^2 - r^2} - c \quad (5.6)$$

where $H$ is the studio length. Likewise, we compute the parallax $P_{13}$ that is caused by cameras $C_1$ and $C_3$, as well as the one $P_{32}$ caused by cameras $C_3$ and $C_2$. The detailed derivation process of all the formulas is described in the appendix.

$$P_{13} = \frac{H \cdot b \cdot c - H \cdot r \cdot \sqrt{4b^2 + c^2 - 4r^2}}{2b^2 - 2r^2} - \frac{c}{2} + \frac{H \cdot r}{\sqrt{b^2 - r^2}} \quad (5.7)$$

$$P_{32} = \frac{H \cdot b \cdot c + H \cdot r \cdot \sqrt{4b^2 + c^2 - 4r^2}}{2b^2 - 2r^2} - \frac{c}{2} - \frac{H \cdot r}{\sqrt{b^2 - r^2}} \quad (5.8)$$

If the object is a plane, projection point $P_{A_3}$ will precisely fall in the middle between $P_{A_1}$ and $P_{A_2}$. For the object with a curved silhouette, the intersection points of its silhouette and the rays from three cameras are separate. The point $P_{A_3}$ will deviate off the center point of $P_{A_1}$ and $P_{A_2}$. So the the parallax values $P_{13}$ and $P_{32}$ are different unless the radius equals zero.

Based on the above formulas, we find the parallax differs with the varying of object radius, its position and camera distance. If the object position and camera distance are fixed, the parallax will increase with the rise of the radius as shown in Formula 5.6. Since the target foreground object is a person, we assume that the radius $r$ varies from 0 to 0.3 meters, the largest diameter of the cylinder is the largest breadth of an adult. The restrictions of object position and camera distance are derived in the next sections.

Figure 5.7: With the radius varying, the parallaxes are changing as well.
5.4. Silhouette error

The difference of camera positions introduces not only parallax, but also silhouette error. When the multiple cameras overlap together and become one, there will be no parallax or silhouette error. This error will result in the change of the object’s silhouette. Moreover, the curved silhouette also causes the separation of the silhouette point in the different views of the multiple cameras. We define the difference of the silhouette point in the different views of the multiple cameras as the **silhouette error**. For instance, Fig. 5.8 shows that the silhouette error caused by cameras $C_1$ and $C_2$ is measured by the difference between the $x$ coordinates of projections of $A_1$ and $A_2$ on the image plane.

![Figure 5.8: The silhouette error caused by cameras $C_1$ and $C_2$](image)

Figure 5.8: The silhouette error caused by cameras $C_1$ and $C_2$ is mathematically measured by the distance between the projection points of $P_{A_1}$ and $P_{A_2}$ onto the image plane.

We wish there is barely no difference of the object silhouette in the three output images of the multiple cameras, which should be accurate at a single-pixel level. So the silhouette error requires to be less than one pixel, which we call the **pixel accuracy constraint**. When the radius becomes largest and model position is fixed, the silhouette error caused by cameras $C_1$ and $C_2$ is the biggest due to the largest camera distance compared to the other ones. So we need to minimize this error caused by cameras $C_1$ and $C_2$, instead of the error caused by cameras $C_1$ and $C_3$ or the one caused by cameras $C_2$ and $C_3$. When the biggest error is minimized and less than one pixel, the smaller ones can be also under the limitation. We assume camera $C_1$ is the origin and formulate the silhouette error in the clip space of camera $C_1$ based on the perspective projection equations:
\[
\text{Error} = f \left( \frac{X_{A2}}{Z_{A2}} - \frac{X_{A1}}{Z_{A1}} \right) = f \left( \frac{r c^2}{2 \sqrt{c^2/4 + b^2 - r^2 (b^2 - r^2)}} \right)
\] (5.9)

where \(X_{A1}, Z_{A1}, X_{A2}\) and \(Z_{A2}\) are the world coordinates of the silhouette points \(A1\) and \(A1\). Parameter \(f\) represents the focal length of the camera. According to the viewport transformation equations, the silhouette error can be represented in pixel coordinates. Based on the previous assumptions (image dimension is 1024 \(\times\) 1024, focal length \(f\) is 50mm, and the largest radius \(r\) equals 0.3m), the pixel accuracy constraint can be expressed in the following inequality:

\[
\text{Error}_{\text{pixel}} = \frac{\text{image\_width} - 1}{2} \times f \times \left( \frac{r c^2}{2 \sqrt{c^2/4 + b^2 - r^2 (b^2 - r^2)}} \right) \leq 1
\] (5.10)

This inequality can help us limit the range of camera distance \(c\) and model position \(b\).

**5.5. Resolution of the object**

Another important element should be taken into account, i.e., the size of the foreground object in an image, which we define as the resolution of the object. This parameter has a contradicting effect. If the object moves far away from the cameras, the silhouette error is reduced, but the resolution of the object is lowered. For instance, Fig. 5.9 shows that given the leftmost silhouette point \(A_1\) and rightmost point \(B_1\) viewed from camera \(C_1\), the resolution of the object is measured by the difference between the \(x\) coordinates of projections of \(A_1\) and \(B_1\) in the clip space of camera \(C_1\). The resolution of the object need to be within a certain threshold to get a high quality output image.

The resolution of the object in the view of camera \(C_1\) is also determined by parameters \(b, c\) and \(r\) as shown in the following formula:

\[
\text{Resolution} = f \left( \frac{X_{B1}}{Z_{B1}} - \frac{X_{A1}}{Z_{A1}} \right) = f \left( \frac{2r \sqrt{c^2/4 + b^2 - r^2}}{b^2 - r^2} \right)
\] (5.11)

where \(X_{B1}, Z_{B1}, X_{A1}\) and \(X_{A1}\) are the world coordinates of silhouette points \(B_1\) and \(A_1\).

When we see rich gallery of portrait photographies, the model usually occupies at least 25% of the whole photo. So we restrain the proportion that
Figure 5.9: The resolution of the object in the image taken by camera $C_1$ is mathematically measured by the distance between the projection points of $A_1$ and $B_1$ in the clip space.

The resolution of the object takes up in the photos by the following inequality:

$$\text{Resolution}_{\text{pixel}} = \frac{\text{image}_width - 1}{2} \ast f \left( \frac{2r\sqrt{c^2/4 + b^2 - r^2}}{b^2 - r^2} \right)$$

$$\geq 25\% \ast (\text{image}_width - 1) \tag{5.12}$$

The resolution of the object restriction helps for further limiting the range of camera distance $c$ and model position $b$.

5.6. Background pattern design

Based on the parallax formulas, the parallax will increase with the rise of object radius if the camera distance and model position remain unchanged. When radius is 0, the parallaxes $P_{12}$, $P_{13}$ and $P_{32}$ all get to the smallest values, and $P_{13}$ and $P_{32}$ are equal to each other. We minimize the parallax $P_{13}$ or $P_{32}$ when the radius equals 0 and other variables are limited by the following constraints, which are the restrictions of camera distance, model position, silhouette error and the proportion that the object takes up in the image:

$$\min_{b,c} \{ P_{13} \} \text{ under } \left\{ \begin{array}{l} \text{Camera Distance } c \geq \text{Minimal Value} \\ \text{Model Position } b \geq \text{Largest Model Radius } r \\ \text{Error}_{\text{pixel}} \leq 1 \\ \text{Resolution}_{\text{pixel}} \geq 25\% \ast 1024 \end{array} \right\}$$

We then assign the optimized minimal $P_{13}$ value to the pattern frequency $d$, which occurs at the new upper bound $UB_b$ of model position $b$ and new lower bound $LB_c$ of camera distance $c$. When camera distance $c$ becomes
larger than $LB_c$, radius is greater than 0 or model position is $b$ smaller than $UB_b$, the parallax values will always be larger than pattern frequency $d$.

5.7. Cameras distance and model location optimization

Based on the pattern frequency $d$, the parallax $P_{12}$ needs to be within the $[2d, 3d]$ in order to make the three projection points locate in three adjacent stripes of the background pattern. Fig. 5.10 shows the lower limit that the parallax value $P_{12}$ equals $2d$ when the radius is 0. The three projection points of the silhouette fall on the boundaries of the two adjacent stripes. In this case, the third camera can also provide effective assistance during segmentation. Because the left or right side of the silhouette just happen to be one color in the view of camera $C_1$, the same side of the silhouette is exactly the opposite color in the view of camera $C_3$. Under the following constraints, we maximize the parallax $P_{12}$ with the largest radius $r$.

$$
\max_{b, c} \{ P_{12} \} \quad \underbrace{\begin{array}{l}
P_{12} = \frac{H_c b}{b^2 - r^2} - c \leq 3d \\
P_{12} = \frac{H_c b}{b^2 - r^2} - c \geq 2d
\end{array}}_{\text{subject to}}
$$

The maximal value $P_{12}$ occurs at the new lower bound $LB_b$ of the model position $b$ and new upper bound $UB_c$ of the camera distance $c$. When the camera distance $c$ becomes shorter than $UB_c$, the radius is smaller than 0.3m or the model position $b$ is larger than $LB_b$, $P_{12}$ will always be smaller than $3d$. So the ranges of the camera distance and model position are:

$$
\begin{cases}
LB_c \leq c \leq UB_c \\
LB_b \leq b \leq UB_b
\end{cases}
$$

Any combination of $b$ and $c$ can work as long as they are within the range.

Figure 5.10: Lower limit. Parallax value $P_{13}$ is equal to $P_{32}$. Three projection points of the silhouette fall on the boundaries of the two adjacent stripes.
In this chapter, we present some insights on fore- and background segmentation using multiple images taken from the three cameras. The segmentation is based on the observation that foreground object position will shift viewed from multiple cameras at different locations. The position change combining the special-designed background pattern is used to assist the segmentation process and produce the final results, which can handle color camouflage problem. The segmentation model includes two steps: background subtraction and then foreground alignment.

6.1. Background subtraction

Background subtraction is an important technique in the fields of image processing and computer vision wherein the foreground object is detected from the difference between the entire image and its background image. Then the foreground object is extracted for further processing. In this thesis, the image's region of interest is human in its foreground. Nevertheless, color camouflage is a great challenge in background subtraction, resulting in holes in the detected foreground image when parts of foreground object share the similar color with the background. We aim to address this issue. The input photos taken from the triple cameras are available to solve the color camouflage problem.

As shown in Fig. 6.1, once the background pattern and camera positions are fixed, we can separately take a shot of the background at first, which is stationary and unchangeable in every view of the triple cameras. After the model stands at the designated location, we can take photos of the model using the triple cameras. These three obtained images are the ones that...
need to be post-processed. Let $I^B_{\text{left}}$, $I^B_{\text{mid}}$, and $I^B_{\text{right}}$ be the known background images for each view of the triple cameras, also called the reference images, and $I^L$, $I^M$, and $I^R$ be the obtained entire images with the foreground object against the background. Taking one pair images $I^B_{\text{left}}$ and $I^L$ as an example, the simplest way to implement the background subtraction is to compare the image $I^L$ with the background image $I^B_{\text{left}}$ pixel by pixel. Since we already have the background image $I^B_{\text{left}}$, we can easily segment out the foreground object by using image subtraction technique, which means for each pixels in the image $I^L$, we can simply subtract the pixel value denoted by $V[I^L]$ with the corresponding pixels at the same position on the background image $I^B_{\text{left}}$. Therefore, the foreground image of $I^L$ can be formulated as described in the following mathematical equation:

$$V[I^F_{\text{left}}] = V[I^L] - V[I^B_{\text{left}}] \quad (6.1)$$

where $I^F_{\text{left}}$ represents the foreground of the input image $I^L$. The output difference image $I^F_{\text{left}}$ shows the intensity or color information for the pixels that have changed in the two images.

![Figure 6.1: Several input images are available for assisting image segmentation process. (a) Image taken from the left camera $C_1$ and its corresponding background image $I^B_{\text{left}}$ on the top row. (b) Image $I^M$ taken from the middle camera $C_3$ and its corresponding background image $I^B_{\text{mid}}$ on the top row. (c) Image $I^R$ taken from the right camera $C_2$ and its corresponding background image $I^B_{\text{right}}$ on the top row.](image-url)
6.2. Foreground alignment

From the above Equation 6.1, we obtain three extracted foreground images that are named $I_{left}^F$, $I_{mid}^F$ and $I_{right}^F$. When the fore- and background share the similar color, the $I_{left}^F$, $I_{mid}^F$ and $I_{right}^F$ will not be complete individually. For instance, if the model wears a white shirt and some parts of the white shirt fall on the white stripes, then these parts will be subtracted and regarded as background. Fortunately, we have other views to compensate. So the final segmentation result is the combination of all the extracted foreground images. However, the extracted foreground images cannot be directly composited together due to the position shift in the different images. We need to align the images.

The aim of alignment is to compare the shapes and sizes of two models, then obtain a similar placement by optically translating, rotating and uniformly scaling the objects. In this thesis, we can keep the models unmodified and avoid the deformation with respect to rotating and scaling. More importantly, the silhouette error is always enforced to be less than one pixel, which means the human eyes can barely see the difference of the model silhouette in the different views of the multiple cameras. The error can be negligible, which is the key in ensuring a good match. We manually translate and align the foreground images for now. In the future, the horizontal translation distance will be determined to achieve precise and automatic alignment.
This chapter discusses the implementation of the system. We provide the technical information about the system, including the system design decisions. Then we outline the features of our system, showing the various interactions. In addition, the entire code of the system has been verified, checked and tested by several people in different ways. This helped to ensure that the implementation is clear, understandable, and robust.

7.1. System design
Our visualization system has been developed using C++, Qt libraries for the User Interface (UI), and OpenGL for modeling 3D objects. The aim of our system is to give the user an intuitive perception and provide the essential information about the appropriate background pattern, camera location and model position in the shooting studio in order to assist the afterward segmentation process. It can also help us verify and test our algorithm under various situations.

7.1.1. Integrated Development Environment
For the Integrated Development Environment (IDE), we use Visual Studio 2013 to develop, debug and test our system due to its versatility, easy extension and the fact that it allows the plugging of functionality.

7.1.2. Qt framework
Our system uses the Qt framework to design the user interface as shown in Fig. 7.1. Qt is a mature application and UI framework. The notable advantages about Qt are open-source and cross-platform. Moreover, its documentation is of exceptionally good quality. Therefore, Qt enjoys a vast com-
munity of developers who target the platform and offer support to others. Our UI contains the front views of the triple cameras (Fig. 7.1 (a), (b) and (c)), a top view for observing the silhouette error (Fig. 7.1 (d)) and a control panel (Fig. 7.1 (e)). The control panel includes:

- A. camera position controller for the user to adjust the distance between the cameras.

- B. upper or lower limit buttons that shows the range of the model position. If the model goes beyond the limit, the silhouette can not be guaranteed to fall on different colors of the background pattern on the wall.

- C. object parameter settings for the user to change the object radius and location.

- D. studio size editor, which is to edit the studio dimension to calculate the background pattern frequency, camera distance and model position for the specific studio.

Figure 7.1: User Interface of our system with four viewers. (a) Left view of the left camera $C_1$. (b) Middle view of the middle camera $C_3$. (c) Right view of the right camera $C_2$. (d) Top view of the cylinder. (e) Control panel.
7.1.3. Open Graphics Library
OpenGL (Open Graphics Library) is a library that we use in our system for developing 3D graphics, such as the 3D cylinder model. It has gained a lot of popularity and been adapted to applications since it was introduced. The advantage is that OpenGL is well documented and accepted as a standard in many different contexts.

Another C++ library libQGLViewer based on Qt that eases the creation of OpenGL 3D viewers is also used in our system. It provides some of the typical 3D viewer functionalities, such as the object selection, stereo display, multiple views and much more. We use it to create four viewers and display the same scene.

7.2. System features
7.2.1. Multiple views of the same model
Our system supports different views of the same model, including the classic front and top views. The top view can clearly show the relationship between the silhouette error and the camera and model positions. The front views are more prone to display that the silhouette of the object falls on different stripes of the background pattern in the different views of the triple cameras.

7.2.2. Simultaneous views
The system allows multiple views to be concurrently visible. They can display the same model in the different views of the three cameras, which is very useful for us to inspect all the performances in different views under various situations in order to gain a better understanding of it. For example, when the cylinder moves towards the cameras, the views change simultaneously to display what background pattern the silhouette of the cylinder falls on in every view. So we can easily see the influence of the object position and camera distance.

7.2.3. Customization
Our system is customizable. The users are able to freely enter the parameters, such as the studio size. Then the system provides the appropriate background pattern frequency and the range of camera and model positions based on their shooting studio size. Table 7.1 shows some examples. If the studio size cannot fulfill the constraints defined in our method, there will be no outputs.
### Table 7.1: Customization Examples

<table>
<thead>
<tr>
<th>Input Size</th>
<th>Radius</th>
<th>Output frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 × 4.5</td>
<td>0.3m</td>
<td>0.07m</td>
</tr>
<tr>
<td>2 × 1.5</td>
<td>0.325m</td>
<td>0.012m</td>
</tr>
<tr>
<td>1 × 0.5</td>
<td>0.3m</td>
<td>None</td>
</tr>
</tbody>
</table>

#### 7.2.4. Direct manipulation
The users can easily manipulate the object and model positions by the control panel. Interactions are possible using the mouse or the keyboard, such as pressing the buttons to control the object position, entering the parameters, invoking menu commands, etc.

#### 7.2.5. Two dimensional canvas
The 3D object - cylinder can be positioned on a two dimensional canvas in our system. Since we can not see the camera position in 3D views, a pop-up panel is designed to display the 2D top view of our system setup as shown in Fig. 5.10, including the background, the cameras and the object in-between.
Results

Our method is integrated in a complete pipeline for the photographers to setup their shooting studio and accurately separate the fore- and background. The workflow is very simple for the users. What they need to do is to input the dimension of their studio. Then based on the algorithm, our system will provide the suggestions about background pattern, camera distance and model location. The images captured in our designed configuration can help the photographers precisely remove the background without considering the color camouflage problem.

We conducted experiments to test our algorithm on several datasets. It works for synthetic data (Fig. 8.3 and Fig. 8.4), but also real photographs (Fig. 8.5). For the synthetic images, we model the 3D studio scene with dimension 6 meters × 4.5 meters in Blender. We assume the largest radius of a human model is 0.3 meter, i.e., \( r \) is equal to 0.3. Based on our algorithm, we design the background pattern according to the input studio size, which is equal to 0.07 meter calculated by our system. Then the camera distance and model position is optimized under the second constraint in our system, which is the parallax value \( \in [2d, 3d] \). We place our model and cameras within the optimized range. We obtain three background images with the special pattern in the different views of the tripe cameras as shown in the first row of Fig. 8.3. Three entire photos with the foreground model are captured by the triple camera-setup as shown in the second row of Fig. 8.3. We also test our method on the model whose clothes are very similar with the background pattern as shown in Fig. 8.4. Moreover, we place our model and cameras out of the optimized range and the obtained images are shown in Fig. 8.6. For the real photos, we setup a mini studio, whose dimension 0.2
meter × 0.15 meter is one-tenth of the real one. The largest radius of our model named Meowth is 0.0325 meter, which is also about one-tenth of the largest girth of a real human model. Our system suggested us to paint the background with the special pattern whose frequency is 0.012 meter. We place our model and cameras within the optimized range. Then the background images and the captured entire images with the foreground model are obtained as shown in the top two rows of Fig. 8.5. Finally we use Matlab to subtract the background from the entire images and manually align the foreground objects together.

The final results of the Example 1, 2 and 3 are shown in Fig. 8.1. We place the model and cameras within the optimized range calculated by our system in these three experiments (Fig. 8.3, Fig. 8.4 and Fig. 8.5). When we use one image taken from a single camera, the result image is incomplete. Some parts of the model are missing because they share similar colors with the background image. When we combine the three extracted foreground images taken from the three cameras, the result gets better and the foreground object is complete. However, there is a lot of noisy in the real photos due to the lighting conditions and a small shift between the background images and the captured photos. In the synthetic images, the noisy is caused because we chose a low threshold for converting the color images to the binary images.

Additionally, we place the model and cameras out of the optimized range calculated by our system in the fourth example (Fig. 8.6). The final result is shown in Fig. 8.2. We cannot ensure all the silhouette points project on the different colors of the background pattern outside the optimized range. There may be some parts that are missing in all the views of the triple cam-

![Figure 8.1](image-url): Final results. (a) the final result of Example 1 (Fig. 8.3). (b): the final result of Example 2 (Fig. 8.4). (c): the final result of Example 3 (Fig. 8.5).
eras, because the projections of these parts all fall on the same color of the background pattern via the triple cameras. None of the views of the triple cameras can help in assisting the image segmentation process and resolving the color camouflage problem.

Figure 8.2: Failure results (Fig. 8.6). The pants are incomplete.
Figure 8.3: Example 1. Some parts of the foreground model share the similar color with the background. Top row: background images. Second row: the entire images taken from the triple cameras. Third row: background subtraction. Fourth row: contour detection. (a) Left view. (b) Middle view. (c) Right view.
Figure 8.4: Example 2. The model wears the shirt with checkerboard pattern and the pants with stripes pattern.
Figure 8.5: Example 3. Real photos.
Figure 8.6: Example 4. Failure result.
Limitations and Discussion

Although the end results show that our method can deal with the color camouflage and avoid color bleeding in the segmentation process, our system still faces several limitations:

9.1. Illumination conditions
One limitation is the assumption that illumination would not be affected for one image pair with or without the foreground model. Because the separation process is to compare the image pair and find their difference pixel by pixel. Hence, the illumination level and direction must be the same for the image pair, then the background can stay same. However, illumination needs to adjust and change to avoid the shadows of the foreground model. So the background in the image pair can not be constant or same in real applications. We could improve the segmentation algorithm by only using one captured image instead of the image pair.

The other restriction is that we found the shadows caused wrong segmentation results. Although we pull the model 3 feet away from the background wall to prevent shadows and allow for easy lighting, the cast shadow is inevitable due to the varying shapes of the human body and inappropriately lighting. Especially there are many shadow areas near the model shoes. One possibility would be to use the shadow removal techniques to remove the shadows from the captured images first, and then subtract the background from the shadow-free images.
9.2. Multiple images
In general, approaches that use multiple images tend to produce better and more robust results. Our approach assumes that the foreground model in the three captured images stays in the same pose. However, to be practical, it is difficult to align the extracted foreground image in cases where the model moves or the fine structures move, as in the case of hair or a dress photographed during a wind simulation. One possible solution is to customize the capture system for simultaneous three cameras capture, for example, using a triple-camera system controlled by one button. This system could also be electronically set up to capture three images at the same time. Three cameras can be triggered to take photos simultaneous once the control button is activated.

9.3. Multiple cameras
This multi-camera method can achieve high accuracy, but it is repeatable and monotonous. Three images of the foreground object against a same background wall need to be shot using three cameras that are located at different positions. Then the same segmentation process has to be done for three times, followed by the alignment procedure. Whenever this method is applicable, however, it is powerful and accurate.
In this thesis project, we propose a novel method for the accurate fore- and background segmentation in the context of a shooting studio. We take advantage of the parallax introduced by the three cameras, combining the special-designed background pattern, to assist the image segmentation and avoid the color camouflage and bleeding problems. Under the constraints - achieving pixel accuracy without sacrificing the resolution of the object, the background pattern frequency is determined by minimizing the parallax $P_{13}$. Another constraint is applied to the parallax $P_{12}$ that $P_{12}$ needs to stay in the range $[2*d, 3*d]$, which aims to ensure the silhouette of the foreground object falls on different colors of the background pattern on the wall. Furthermore, the second constraint also helps us to optimize the camera distance and model position, which is the essential information for the photographers to setup their shooting studios. Additionally, we also implement a visualization system, which is developed to facilitate the photographers to setup their shooting studios by providing the appropriate information about the background pattern, the camera distance and model position based on their shooting space.

We showed segmentation results of the images taken from a virtual studio, including the synthetic images and real photos. The experiment results show that our solution enables users to generate the segmentation results precisely and effectively avoid the color camouflage and bleeding problems.

The future work can be conducted in four aspects:

- We will improve the segmentation model to deal with the illumination
issues by using one image instead of image pair to extract the foreground object.

- We will calculate the horizontal translation distance to automatically and precisely align the foreground images.

- We will also investigate how to take advantage of the silhouette error to detect the silhouette of the foreground object and assist the segmentation process or enhance the edge features. This idea, if further developed, may be also useful for the recognition tasks.

- We will further improve the system - replacing the cylinder model with the human model, which is more intuitive and easy to be understood. We also plan to integrate the segmentation and the optimization processes into one platform.
Bibliography


matting. In *ACM Transactions on Graphics (ToG)*, volume 23, pages

[55] Jian Sun, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. Flash mat-
ting. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 772–

blue-screen-vs-green-screen.


[58] Eduard Vazquez, Joost van de Weijer, and Ramon Baldrich. Histogram-
based image segmentation in the presence of shadows and highlights.

ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages

[60] Wenyi Wang. *Robust Chroma Keying System Based on Human Vi-
sual Perception and Statistical Color Models*. PhD thesis, Université
d’Ottawa/University of Ottawa, 2016.

[61] Yucheng Wang and Yunde Jia. A fusion framework of stereo vision and
kinect for high-quality dense depth maps. In *Asian Conference on Com-

[62] Yair Weiss. Deriving intrinsic images from image sequences. In *Com-

westcottu.com/the-essential-guide-to-backdrops.


[65] Tai-Pang Wu, Chi-Keung Tang, Michael S Brown, and Heung-Yeung
Shum. Natural shadow matting. *ACM Transactions on Graphics (TOG)*,


Formulas Derivation of Silhouette Error and Resolution

Our algorithm analysis starts with the simplifying assumption that the target object is a cylinder with varying radius as shown in Figure 5.1, which is a 2D top view of our system. We denote the angles between \( \angle OC_1C_2 = \alpha \), \( \angle OC_1A_1 = \beta \), \( \angle OC_2C_1 = \alpha \), \( \angle OC_2A_2 = \beta \). Camera \( C_1 \) is the origin. We can have the following world coordinates:

\[
A_1 = (C_1A_1 \cos(\alpha + \beta), 0, C_1A_1 \sin(\alpha + \beta)) \quad (A.1)
\]

\[
A_2 = (c - C_2A_2 \cos(\alpha - \beta), 0, C_2A_2 \sin(\alpha - \beta)) \quad (A.2)
\]

\[
B_1 = (C_1A_1 \cos(\alpha - \beta), 0, C_1A_1 \sin(\alpha - \beta)) \quad (A.3)
\]

\[
B_2 = (c - C_2A_2 \cos(\alpha + \beta), 0, C_2A_2 \sin(\alpha + \beta)) \quad (A.4)
\]

We assume \( C_1A_1 \) equals to \( m_1 \). \( m_1 \) actually equals to \( \sqrt{c^2/4 + b^2} - r \). The length value \( C_2A_2 \) also equals to \( m_1 \). In addition, we used many trigonometric functions during the derivation:

\[
\sin \alpha = \frac{b}{\sqrt{c^2/4 + b^2}} \quad (A.5)
\]

\[
\cos \alpha = \frac{c}{2\sqrt{c^2/4 + b^2}} \quad (A.6)
\]

65
\[
\sin \beta = \frac{r}{\sqrt{c^2/4 + b^2}} \quad (A.7)
\]
\[
\cos \beta = \frac{m_1}{\sqrt{c^2/4 + b^2}} \quad (A.8)
\]
\[
\cot(\alpha + \beta) = \frac{\cot \alpha \cdot \cot \beta - 1}{\cot \beta + \cot \alpha} \quad (A.9)
\]
\[
\cot(\alpha - \beta) = \frac{\cot \alpha \cdot \cot \beta + 1}{\cot \beta - \cot \alpha} \quad (A.10)
\]

Based on the above functions, the silhouette error caused by cameras \( C_1 \) and \( C_2 \) can be calculated by the following equation:

\[
\text{Error} = f \left( \frac{X_{A2}}{Z_{A2}} - \frac{X_{A1}}{Z_{A1}} \right) = \]
\[
f \left( \frac{c}{m_1 \sin(\alpha - \beta)} - \cot(\alpha - \beta) - \cot(\alpha + \beta) \right) = \]
\[
f \left( \frac{c \cdot r(c - a)}{m_1 \cdot b^2 - r^2} \right) = f \left( \frac{r c^2}{2m_1(b^2 - r^2)} \right) = \]
\[
f \left( \frac{r c^2}{2 \sqrt{c^2/4 + b^2 - r^2}(b^2 - r^2)} \right) \quad (A.11)
\]

, where \( f \) is a constant, representing focal length of the camera.

The resolution of the object in the view of camera \( C_1 \) is calculated as follows:

\[
\text{Resolution} = \]
\[
f \left( \frac{X_{B1}}{Z_{B1}} - \frac{X_{A1}}{Z_{A1}} \right) = f \left( \frac{2m_1 r}{b^2 - r^2} \right) = \]
\[
f \left( \frac{2r \sqrt{c^2/4 + b^2 - r^2}}{b^2 - r^2} \right) \quad (A.12)
\]
Parallax Formula Derivation

In order to find the optimal background pattern frequency, we need to analyze the parallax, which is measured by the distance between the projection points of the silhouette on the background wall via the three cameras.

We assume the studio length is a constant $H$, The $X$ coordinates of projection points $P_{A_1}$, $P_{A_2}$ and $P_{A_3}$ on the background wall are named $X_1$, $X_2$, $X_3$ in this appendix, respectively. The $Z$ coordinates of these three projected points are all equals to $H$. As we know, the distance from the line $C_1P_{A_1}$ to point $O$ is $r$, the radius of the cylinder. So are the distances between the line $C_2P_{A_2}$ and point $O$, and the line $C_3P_{A_3}$ and point $O$. The equations of lines $C_1P_{A_1}$, $C_2P_{A_2}$ and $C_3P_{A_3}$ are shown in the following:

1. $C_1P_{A_1} : H \cdot X - X_1 \cdot Z = 0 \quad \text{(B.1)}$
2. $C_2P_{A_2} : H \cdot X + (c - X_2) \cdot Z - c \cdot H = 0 \quad \text{(B.2)}$
3. $C_3P_{A_3} : 2H \cdot X + (c - 2X_3) \cdot Z - c \cdot H = 0 \quad \text{(B.3)}$

Then the distances from the point $O$ to these three lines are:

1. $O \leftrightarrow C_1P_{A_1} : \frac{|H \cdot \frac{c}{2} - X_1 \cdot b|}{\sqrt{H^2 + X_1^2}} = r \quad \text{(B.4)}$
2. $O \leftrightarrow C_2P_{A_2} : \frac{|H \cdot \frac{c}{2} + (c - X_2) \cdot b - c \cdot H|}{\sqrt{H^2 + (c - X_2)^2}} = r \quad \text{(B.5)}$
**B. Parallax Formula Derivation**

\[
O \leftrightarrow C_3 P_{A_3} : \frac{|2H \cdot \frac{c}{2} + (c - 2X_3) \cdot b - c \cdot H|}{\sqrt{(2H)^2 + (c - 2X_3)^2}} = r \quad \text{(B.6)}
\]

Therefore, we can get the X coordinates of \(P_{A_1}\), \(P_{A_2}\) and \(P_{A_3}\) are:

\[
X_1 = \frac{H \cdot b \cdot c - H \cdot r \cdot \sqrt{4b^2 + c^2 - 4r^2}}{2b^2 - 2r^2} \quad \text{(B.7)}
\]

\[
X_2 = c - \frac{H \cdot b \cdot c + H \cdot r \cdot \sqrt{4b^2 + c^2 - 4r^2}}{2b^2 - 2r^2} \quad \text{(B.8)}
\]

\[
X_3 = \frac{c}{2} - \frac{H \cdot r}{\sqrt{b^2 - r^2}} \quad \text{(B.9)}
\]

So we can get the distances between X coordinates of \(P_{A_1}\) and \(P_{A_2}\) as \(P_{12}\), \(P_{A_1}\) and \(P_{A_3}\) as \(P_{13}\), \(P_{A_3}\) and \(P_{A_2}\) as \(P_{32}\)

\[
P_{12} = X_1 - X_2 = \frac{H \cdot c \cdot b}{b^2 - r^2} - c \quad \text{(B.10)}
\]

\[
P_{13} = X_1 - X_3 = \frac{H \cdot b \cdot c - H \cdot r \cdot \sqrt{4b^2 + c^2 - 4r^2}}{2b^2 - 2r^2} - \frac{c}{2} + \frac{H \cdot r}{\sqrt{b^2 - r^2}} \quad \text{(B.11)}
\]

\[
P_{32} = X_3 - X_2 = \frac{H \cdot b \cdot c + H \cdot r \cdot \sqrt{4b^2 + c^2 - 4r^2}}{2b^2 - 2r^2} - \frac{c}{2} - \frac{H \cdot r}{\sqrt{b^2 - r^2}} \quad \text{(B.12)}
\]