

# **Segmentation of RGB-D Data Using RGB-Based Background Subtractors: Two Proposed Methods for Robust Segmentation of Camouflaged Objects**

JACOB FIOLA



University of Colorado  
Boulder

Supervisor: Alessandro Roncone, PhD  
Associate Supervisor: Anuj Pasricha, MS

A thesis submitted in fulfilment of  
the requirements for the degree of  
Bachelors of Science in Computer Science

College of Engineering & Applied Science  
University of Colorado  
Boulder

5 May 2020

## **Abstract**

This thesis addresses the problem of segmenting camouflaged objects of interest using background subtraction techniques. If this problem can be resolved, less post-processing will be necessary for the optimal segmentation of camouflaged objects of interest.

If a camouflaged object enters the scene, most traditional RGB based background subtractors will fail to create an accurate segmentation of the camouflaged object. This is because there are little to no RGB differences detected. Likewise, if a very flat object is placed on a surface in a scene, most depth based background subtractors will fail to create an accurate segmentation of the flat object. This is because there are little to no depth differences detected. This thesis proposes and evaluates RGB Union + Simple Depth Subtraction (RUSDS), a method for modeling the location of objects using a combination of RGB-based and depth-based background subtractions.

This thesis also proposes evaluates a method called RGB-based Background Subtraction Using Depth Colormap Input (RBSUDCI), which allows any existing RGB-based background subtraction algorithm to use depth value colormaps as an input instead of a traditional RGB image. Both qualitative and quantitative results suggest that RBSUDCI is superior to any RGB-only background subtraction algorithm, and that RUSDS can compete with the performance of the algorithms which utilized the RBSUDCI technique. All algorithms are quantitatively evaluated by calculating their respective IOU's to the ground truth.

## **Acknowledgements**

I cannot express enough thanks to my research advisors for their unrelenting support and encouragement: Dr. Alessandro Roncone, my thesis supervisor; and Mr. Anuj Pasricha, my associate thesis supervisor. I offer my lifetime appreciation for the opportunities provided by these two extraordinary men.

My completion of this project could also not have been accomplished without the support of my research committee: Dr. Chris Heckman, Dr. Bradley Hayes, and Dr. Alessandro Roncone.

Finally, to my caring, loving, and supportive family who put up with me blabbering away about object segmentation. Especially, to my sister, Emily Fiola, who kept me motivated to write a solid thesis.

## Contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Contents</b>	<b>iv</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
<b>Chapter 2 Related Work</b>	<b>3</b>
2.1 Adaptive Gaussian Mixture Models (MOG1) .....	3
2.2 Improved GMMs (MOG2) .....	3
2.3 Local SVD Binary Pattern (LSBP) .....	4
2.4 GSoC .....	4
<b>Chapter 3 Methods</b>	<b>5</b>
3.1 RBSUDCI: RGB-based Background Subtraction Using Depth Colormap Input	5
3.2 RUSDS: RGB Union + Simple Depth Subtraction .....	7
3.2.1 Calibration .....	7
3.2.2 Simple Depth Subtraction .....	8
3.2.3 RGB-Union based subtraction .....	8
3.2.4 Morphology .....	9
3.2.5 Combination .....	9
<b>Chapter 4 Results</b>	<b>10</b>
4.1 Qualitative: Algorithm output images .....	11
4.2 Quantitative: IOU performance comparison .....	16
<b>Chapter 5 Conclusion</b>	<b>18</b>
5.1 Future outlook .....	18
<b>Bibliography</b>	<b>19</b>

## CHAPTER 1

### Introduction

---

To interpret an image, one must first isolate objects within the image and find relations among them. This process of object separation is referred to as image segmentation [1]. The reliable segmentation of objects is a basic, yet critical requirement for many computer vision systems being used in the world today. For instance, autonomous robots working in factory assembly lines must determine the presence of objects in real-time [2]. Self-driving cars need highly reliable segmentations of objects (e.g. lane markings, other cars, pedestrians) within the surrounding environment in order to avoid crashing into something [3]. Segmentation is also key to more high-level, feature-extracting computer vision techniques. For example, if the system obtains a binary mask of an object (a segmentation), then that mask can be used as an input to deep neural networks which have the potential to classify the object [4]. All in all, segmentation is very commonly utilized in the processing pipeline of many computer vision systems.

Background subtraction is a very commonly used segmentation technique [5, 6, 7]. Background subtraction involves comparing a current image with a base image where there are no objects of interest in the scene. The portions of the current image where there is a significant difference from the base image indicate the location of the objects of interest. Although many different novel background subtraction solutions have been proposed, the accuracy of their produced segmentations can be underwhelming when confronted with various problem phenomena, including:

- (1) Objects blending into the background (camouflage)
- (2) Changes in lighting
- (3) Shadows
- (4) Moving objects
- (5) Dynamic backgrounds
- (6) High-Traffic areas

Some background subtraction techniques are robust to specific problems. For example, the Local SVD Binary Pattern (LSBP) algorithm is relatively robust to changes in lighting and shadows [8]. Unfortunately, there is no background subtraction technique that can reliably deal with all of the problem phenomena above.

Some of these problems can be virtually eliminated via a wide array of computationally expensive post-processing techniques after the background subtraction [9]. Furthermore, there exists a post-processing algorithm that recovers camouflage segmentation errors on

moving people [10]. However, many systems, such as self-driving cars, require real-time segmentation. Such systems are even more heavily reliant upon the initial background subtraction without any post-processing.

If the process of background subtraction was more robust towards one of the previously listed problem cases, less post-processing techniques will be necessary. Therefore, the primary purpose of this research is to improve upon background subtraction's handling of the camouflage problem case. In this paper, the camouflage problem case is drastically suppressed when depth based background subtractions are taken into account.

## Related Work

---

All the sections in this chapter cover various background subtraction algorithms. It is important to keep in mind that although these algorithms were designed to have RGB inputs, one could use a specific depth colormap as the input instead. This process will be further elaborated on in Section 3.1.

### 2.1 Adaptive Gaussian Mixture Models (MOG1)

A widely renowned background subtraction technique is to model each pixel of the background within a video frame by using a Normal, aka Gaussian distribution. This strategy is intrinsic to many other background subtraction algorithms, including [11], which models each pixel using  $K$  Gaussians. Each Gaussian represents a distinct color within the pixel. The more dynamic the background, the higher  $K$  should be. For most environments,  $K$  should be in the range of  $[3, 5]$ . The Gaussians are then ordered based on a fitness value, where the most static Gaussian is first. The first  $B$  Gaussians of a pixel is the estimation of the background, where  $B \leq K$ . The scalar values of  $B$  and  $K$  may change from frame to frame, however, each pixel within one frame has the same  $B$  and  $K$  values.

As new video frames come in, the distributions for each pixel color is updated. Then, any pixel which is 2.5 standard deviations from any its  $B$  distributions is marked as a foreground pixel.

The MOG1 background subtraction method is well suited for the multimodal nature of real-life situations. For example, MOG1 can handle segmentation within an environment containing repetitive background motion, such as swaying leaves. However, the fact that  $B$  and  $K$  are kept constant for each pixel is a limiting factor, since some areas of an image may be more dynamic than other areas.

### 2.2 Improved GMMs (MOG2)

This technique is the implementation of papers [12] and [13]. The main difference between this strategy and the strategy described in Section 2.1 is that  $B$  is now correctly calculated for each pixel, rather than being held constant for each frame.

This simple tweak to the previous section’s algorithm results in dramatically better segmentation [14] when compared to the segmentations produced by the strategy in the previous section.

MOG2 is currently one of the best performing RGB-based background subtraction algorithms[15]. This is because MOG2 is well rounded to problem scenarios such as dynamic backgrounds, repetitive background motion, and lighting changes. However, MOG2 still creates a considerable amount of false positive errors from shadows, while also creating a considerable amount of false negative errors from camouflage. These errors are highlighted in Chapter 3, Table 3.1.

## 2.3 Local SVD Binary Pattern (LSBP)

The LSBP algorithm [8] follows a different, linear algebra based approach to segmentation. Local Singular Value Decompositions (SVDs) are calculated for various square neighborhoods of pixels. These SVDS provide descriptions of the structures present within different areas of an image.

Due to the nature of Eigendecompositions, the coefficients of the singular values (the feature descriptors) in the SVD are highly invariant to changes in lighting on the modeled structures. Therefore, if the singular values for a current neighborhood of pixels does not form a consensus with the feature descriptors of the corresponding background model, it is likely that some of the pixels within that neighborhood were significantly altered in composition, undue to lighting changes. This suggests that those previously mentioned pixels may be elements of the foreground. Because feature descriptors are highly invariant to lighting changes, LSBP is inherently robust to changes in lighting and shadows compared to most other background subtraction techniques.

LSBP is known to especially struggle when presented with dynamic backgrounds. Similar to MOG1 and MOG2, LSBP creates a considerable amount of false negative errors from camouflage. There errors are highlighted in Section 4.1.

## 2.4 GSoC

This algorithm was developed during the Google Summer of Code (GSoC), and did not originate from any paper. It is based off the feature descriptors from LSBP, but utilizes a post-processing method which re-analyzes each pixel independently.

Because this method uses LSBP feature descriptors, it is also robust to lighting and shadows. Just like all previous methods, GSOC creates a considerable amount of false negative errors from camouflage. There errors are highlighted in Section 4.1.



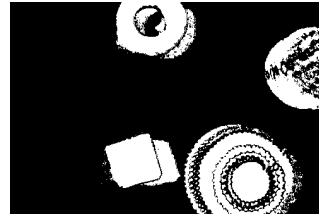
## Methods

---

Even though the methods in Section 2 are well-renowned and widely implemented, they can still create inaccurate segmentations for environments as simple as objects on a table. For example, in Figure 3.1, the MOG2 algorithm performs a qualitatively inaccurate segmentation on RGB data of the previously described environment.



RGB Image



MOG2 segmentation

FIGURE 3.1. The MOG2 algorithm performing segmentation on RGB data of objects on a wooden table

In this MOG2-generated RGB segmentation, shadows create false positives. At the same time, the brown yo-yo creates false negatives because the color of the yo-yo is too similar to the color of the table. The following methods were developed to suppress the problem of false negatives.

### 3.1 RBSUDCI: RGB-based Background Subtraction Using Depth Colormap Input

The false negatives present in the MOG2 segmentation of Figure 3.1 can be attributed to a lack of RGB difference. If depth difference could also be considered, objects such as the yo-yo would be easier to segment, due to depth difference from the background. The amount of false negatives caused by a lack of RGB difference would be significantly lowered. Therefore, the RBSUDCI method was developed to allow existing RGB-based background subtraction algorithms to use RGB-D data.

Every single method described in Chapter 2 can easily translate to use depth colormaps. The outputs of these depth-based segmentations will provide an entirely new perspective on where the objects of interest may or may not be.

Using a traditional RGB input for MOG1 and MOG2 means each pixel Gaussian models a distinct color within the pixel. However, if a depth colormap input is used for MOG1 or MOG2, each pixel Gaussian now models a distinct depth within the pixel. This allows multiple dynamic depths to be accounted for, just like how multiple dynamic colors are being handled.

Likewise, the local SVD decompositions used in the LSBP and GSOC methods will be able to identify invariant properties of the depth image.

When choosing a colormap to use, It is important to choose a high-contrast colormap. The more colors a colormap contains, the higher contrast it is. The highest contrast colormap available in most libraries is the HSV colormap. Consequentially, the HSV colormap was the colormap of choice for this research, as illustrated in Figure 3.2.

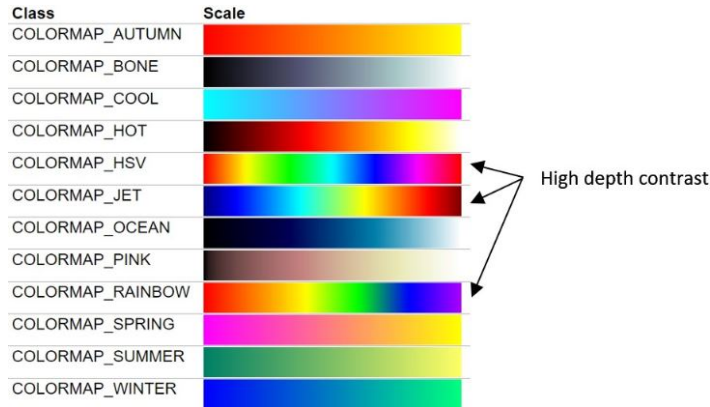


FIGURE 3.2. Optimal colormap choices for depth-based background subtraction

Another colormap parameter to tune is the alpha value, which can be any float  $\in [0, 1]$ . The higher the alpha value, the higher the depth contrast. An alpha value of 0.7 was chosen for this research, as it experimentally yielded the most promising object segmentations. Alpha values greater than 0.7 were creating segmentations which included both the object and the surrounding area. Alpha values less than 0.7 were creating segmentations which did not include the entire object.

Each method given the task of performing background subtractions on the RGB input and the depth colormap. The results of these two segmentations were then combined via union to create a third, combined segmentation. This process is illustrated in Figure 3.3.

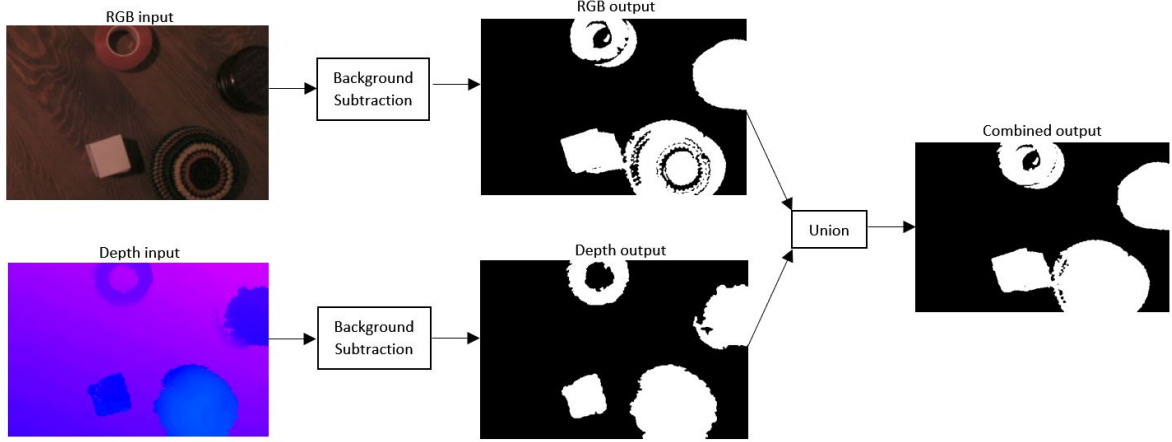


FIGURE 3.3. Any RGB-based background subtraction method can be used to process RGB-D data. Here, the GSOC method performs segmentations on Environment 1. The top process corresponds to segmenting a traditional RGB image input. The bottom process corresponds to RBSUDCI. These two segmentations are then combined via union to create a third, combined segmentation.

## 3.2 RUSDS: RGB Union + Simple Depth Subtraction

The initial goal of this research was to create a tool which enables the creation of simple, yet accurate masks of objects on a flat surface. This prompted the development of RUSDS, a novel background subtraction algorithm based on global thresholds. Because of this, it will also be compared to the rest of the methods described in Chapter 2. After a short calibration period (Section 3.2.1), the RUSDS method performs a Depth subtraction (Section 3.2.2), an RGB subtraction (Section 3.2.3), morphology (Section 3.2.4), and finally a union-combination (Section 3.2.5). This tool has an unsupervised and supervised adaptation as described in Sections 3.2.2 and 3.2.3.

### 3.2.1 Calibration

Before performing any background subtraction, in order to obtain a reliable model of the background, the camera takes in 30 frames of calibration data on the background surface. During this short time of 2 to 4 seconds, the user should not place an object or cast shadows on the surface to be segmented upon. These 30 frames are used to calculate average RGB and depth values for each pixel.

- Per-pixel average color component and depth values are stored as  $C_{\text{initial}}[\text{pixel}][\text{component}]$  and  $D_{\text{initial}}[\text{pixel}] \mid \text{pixel} = (i, j) \in \mathbb{Z}^2, \text{component} \in (R, G, B)$ .
- The average illumination of the image is stored as  $b_{\text{avg}} \in \mathbb{R} \mid 0 \leq b_{\text{avg}} \leq 255$ .
- The average depth of the image in millimeters is stored as  $d_{\text{avg}} \in \mathbb{R} \mid 0 \leq d_{\text{avg}}$ .

- The average HSV values of the image are respectively stored as  $h_{avg}$ ,  $s_{avg}$ , and,  $v_{avg}$  where  $h_{avg} \in \mathbb{R} \mid 0 \leq h_{avg} \leq 179$ ,  $s_{avg} \in \mathbb{R} \mid 0 \leq s_{avg} \leq 255$ ,  $v_{avg} \in \mathbb{R} \mid 0 \leq v_{avg} \leq 255$ .

### 3.2.2 Simple Depth Subtraction

This method was given its name because it utilizes simple global depth difference thresholds for its segmentations. Global thresholding is the one of the simplest techniques used in background subtraction. Although this is a very simple strategy, the depth-based segmentations this method produces are surprisingly accurate.

- As new frames come in, per-pixel depths are stored in  $D_{current}$ .
- A background subtraction is performed:  $D_{differences} = D_{current} - D_{initial}$ .
- Each pixel within  $D_{differences}$  is individually compared to a single scalar global depth threshold,  $t_{depth}$ . Any pixel where  $D_{differences}[\text{pixel}] \geq t_{depth}$  is marked white as a foreground pixel.

The way  $t_{depth}$  is calculated varies from supervised to unsupervised.

**Supervised:** The user manually tunes the value of  $t_{depth}$  to create the best possible segmentation for their specific needs.

**Unsupervised:**  $t_{depth}$  is a function of  $d_{avg}$ , where  $t_{depth} = -0.28d_{avg}^2 + 219.7d_{avg} - 61,208$ . This function is an experimentally driven quadratic regression with the feature being average depth value and the response being optimal  $t_{depth}$  value for that given instance. This regression has an  $R^2$  value of .95, meaning that it explains 95 % of the variance in the experimental data.

### 3.2.3 RGB-Union based subtraction

This method is coined "RGB-Union" because if just one of the red, green, or blue differences of a pixel is higher than the respective threshold, the whole pixel registers as a foreground pixel, no matter the other color component differences.

- As new frames come in, per-pixel colors are stored in  $C_{current}$ .
- A background subtraction is performed:  $C_{Lighter\ differences} = C_{current} - C_{initial}$  and  $C_{Darker\ differences} = (-1) * C_{Lighter\ differences}$
- Each RGB component within  $C_{Lighter\ differences}$  and  $C_{Darker\ differences}$  is then respectively compared to one of the global color thresholds,  $t_{Lighter}$  or  $t_{Darker}$ .
- Any RGB component of a pixel where  $C_{Lighter\ differences}[\text{pixel}][\text{component}] \geq t_{Lighter}$  or  $C_{Darker\ differences}[\text{pixel}][\text{component}] \geq t_{Darker}$  is marked white as a foreground pixel.

The way  $t_{Lighter}$  and  $t_{Darker}$  is calculated varies from supervised to unsupervised.

**Supervised:** The user manually tunes the value of  $t_{Lighter}$  and  $t_{Darker}$  to create the best possible segmentation for their specific needs.

**Unsupervised:**  $t_{Lighter}$  and  $t_{Darker}$  are functions of  $h_{avg}$ ,  $s_{avg}$ ,  $v_{avg}$ , and  $b_{avg}$ , where  $t_{Lighter} = 0.0183h_{avg} + 3.751s_{avg} - 0.5178v_{avg} + 0.2919b_{avg} - 3.56$  and  $t_{Darker} = -0.026h_{avg} + 0.89_{avg} -$

$0.4017v_{\text{avg}} + 0.2572b_{\text{avg}} + 2.33$ . These functions are experimentally driven multi-linear regressions with features of average h,s,v, and luminance. The responses are the optimal  $t_{\text{Lighter}}$  and  $t_{\text{Darker}}$  value for that given instance. Unfortunately, these regressions have respective  $R^2$  values of .48 and .64, which means only 56 percent of the variance in the experimental data is explained. This suggests that the segmentations from the unsupervised method will not be very good. Many different regressions were considered. The two which ended up being chosen had the highest explained variance.

### 3.2.4 Morphology

In order to further smooth out the segmentation and drastically reduce noise, a few morphology operations are performed on the Simple Depth Subtraction and RGB Union binary outputs. These operations can be boiled down to a series of erosions and dilations [16]. Erosion removes noise and small objects so that only substantive objects remain. Dilation makes objects more visible and fills in small holes within objects.

Each segmentation undergoes two morphology operations with a kernel size of (2,2). These two operations are an morphological open (an erosion followed by a dilation) followed by a morphological close (a dilation followed by an erosion). This process is then repeated a second time with a smaller kernel of size (1,1). The larger kernel corresponds with big, coarse operations. The smaller kernel corresponds with finer, polish operations.

### 3.2.5 Combination

After undergoing morphology, the RGB-Union based subtraction and Simple Depth Subtraction binary images are combined via union to create a third, combined segmentation. This process of combination is the exact same combination performed in Section 3.1.

## CHAPTER 4

### Results

---

All of the background subtraction algorithms described in Chapters 2 and 3 were given the task of segmenting objects in five different environments which varied in lighting, background surface, and objects of interest. Some environments included objects with high RGB differences from the background, which in turn allowed for an easy RGB segmentation. Likewise, some environments included objects with high depth differences from the background, which allowed for an easy depth segmentations. These conditions were permuted throughout the five environments as follows:

- (1) Easy RGB and depth segmentation: Colored, highly 3D objects - A cube, a tape roll, a yo-yo, and a hacky sack
- (2) Easy RGB, hard depth segmentation: Colored, highly 2D objects - Hotel key cards
- (3.1) Hard RGB, easy depth segmentation: Camouflaged, highly 3D objects - Paper cubes on a sheet of paper of the same color (Environment 3, case 1)
- (3.2) Dark: Dark objects in a low-illumination environment (Environment 3, case 2)
- (4) Hard RGB and depth segmentation: Camouflaged, highly 2D objects - Shards of paper on a sheet of paper

Some environments highlight the challenge of camouflaged object segmentation more than others. This is to get a feel for the versatility of each algorithm.

The following background subtraction output images are the exact images which were used for IOU scoring in Section 4.2. In this section, the background subtraction methods are organized by column, and the inputs to the methods are organized by row. The first row is the segmentation mask created when the method uses only the RGB input. The second row is segmentation mask created when the method uses only RBSUDCI input, as proposed in Section 3.1. The third row, combined mask, is simply a union of the previous two rows. The first two columns are the supervised and unsupervised versions of RUSDS. The rest of the columns are the methods described in chapter 2.

These experiments were conducted using an Intel RealSense D435i sensor. For each environment, the sensor was placed above a surface, looking straight down. The sensor was approximately 250mm above the table.

## 4.1 Qualitative: Algorithm output images

In environment 1, all RGB-based masks become more accurate once combined with depth-based masks. Namely, the yo-yo and the hackey sack segmentation is drastically improved once depth-based masks are considered. This is because environment 1 represents high RGB and depth difference, which can be created by introducing colored, highly 3D objects into the scene. Therefore, the objects/surface chosen for this environment were a yo-yo, a paper cube, a hackey sack, and a roll of tape on a wooden surface. Depth masks are created in one of two ways: RUSDS uses depth subtraction (Section 3.2.2), while the other four background subtractors consider depth by performing RBSUDCI (Section 3.1).

**Environment 1:** Easy RGB and depth segmentation

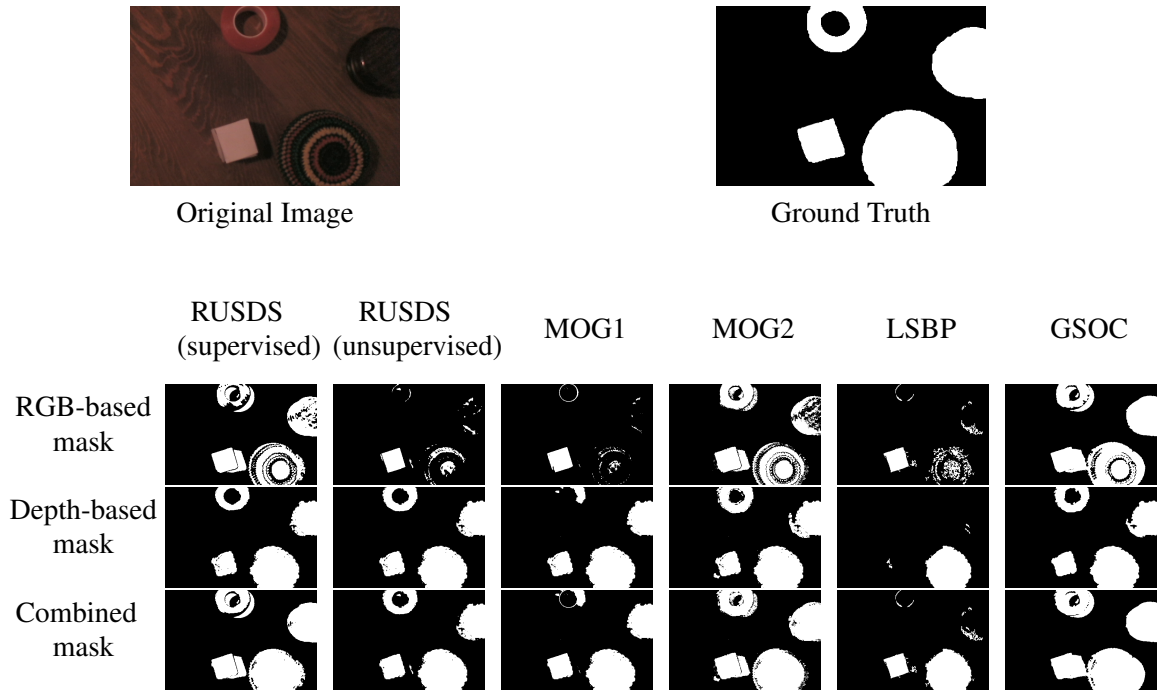


TABLE 4.2. As previously discussed in Chapter 3, the objects in environment 1, especially the yo-yo, are segmented poorly when background subtractors are exposed to only the RGB data (row 1). However, once depth input is taken into account in row 2, every single background subtractor is able to improve upon their initial RGB-only segmentation.

In environment 2, depth-based masks are almost non-existent due to the lack of depth differences present in the scene. This is because environment 2 represents high RGB and low depth difference, which can be created by introducing colored, highly 2D (flat) objects into the scene. Therefore, the objects/surface chosen for this environment were hotel keycards on a wooden surface. Depth masks are created in one of two ways: RUSDS uses depth subtraction (Section 3.2.2), while the other four background subtractors consider depth by performing RBSUDCI (Section 3.1)

**Environment 2:** Easy RGB, hard depth segmentation





















						
	Original Image	Ground Truth				
	RUSDS (supervised)	RUSDS (unsupervised)	MOG1	MOG2	LSBP	GSOC
RGB-based mask						
Depth-based mask						
Combined mask						

TABLE 4.4. The objects in environment 2 are segmented poorly when background subtractors are exposed to only the depth (row 2). This is because the hotel key cards make little to no depth differences in the scene. However, once RGB input is taken into account in row 1, every single background subtractor is able to improve upon their depth-only segmentation. Although the RGB segmentation drastically improves the depth segmentation, it still is not perfect. Most notably, the bottom keycard was inadequately segmented by the unsupervised RUSDS, MOG1, and LSBP methods.



In environment 3.1, RGB-based masks are prone to false positives, while depth-based masks provide a relatively accurate segmentation. This is because environment 3.1 represents low RGB and high depth difference, which can be created by introducing camouflaged, highly 3D objects into the scene. Therefore, the objects/surface chosen for this environment were paper cubes on top of a same-colored sheet of paper. Depth masks are created in one of two ways: RUSDS uses depth subtraction (Section 3.2.2), while the other four background subtractors consider depth by performing RBSUDCI (Section 3.1).

**Environment 3.1:** Hard RGB, easy depth segmentation

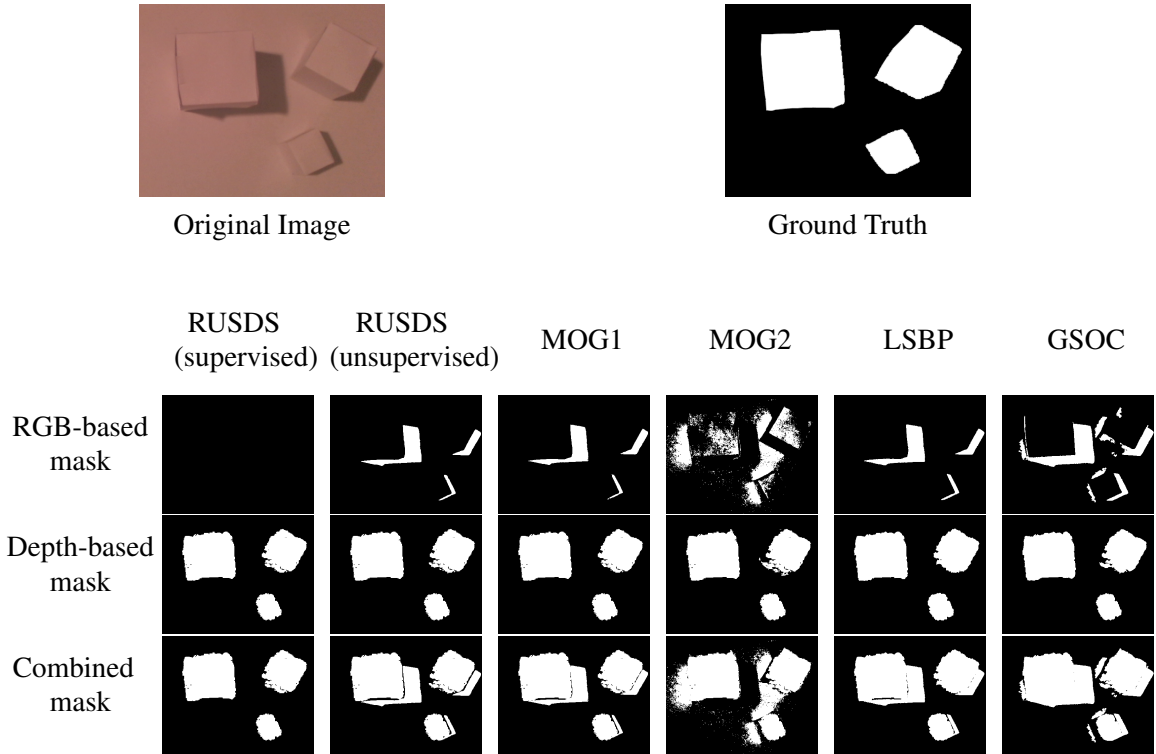


TABLE 4.6. The objects in environment 3 are segmented poorly when background subtractors are exposed to only RGB (row 1). This is because the paper cubes make little to no RGB differences in the scene. However, once depth input is taken into account in row 2, every single background subtractor is able to improve upon their RGB-only segmentation. Although the depth segmentation drastically improves the combined segmentation, it still is not perfect. Most notably, false-positives caused by shadows are present in all combined segmentations except supervised RUSDS. This is because supervised RUSDS allowed for the user to completely ignore the RGB-based segmentation.

In environment 3.2, just like environment 3.1, RGB-based masks are prone to false negatives, while depth-based masks provide a relatively accurate segmentation. This is because environment 3.2 is an extension of environment 3.1, which also represents low RGB and high depth difference. However, these conditions are now being caused by the lack of light in the scene. This environment was created by introducing colored, highly 3D objects into the dark scene. Therefore, the objects/surface chosen for this dark environment were pens and a roll of tape on a wooden surface. Depth masks are created in one of two ways: RUSDS uses depth subtraction (Section 3.2.2), while the other four background subtractors consider depth by performing RBSUDCI (Section 3.1).

**Environment 3.2:** Dark room

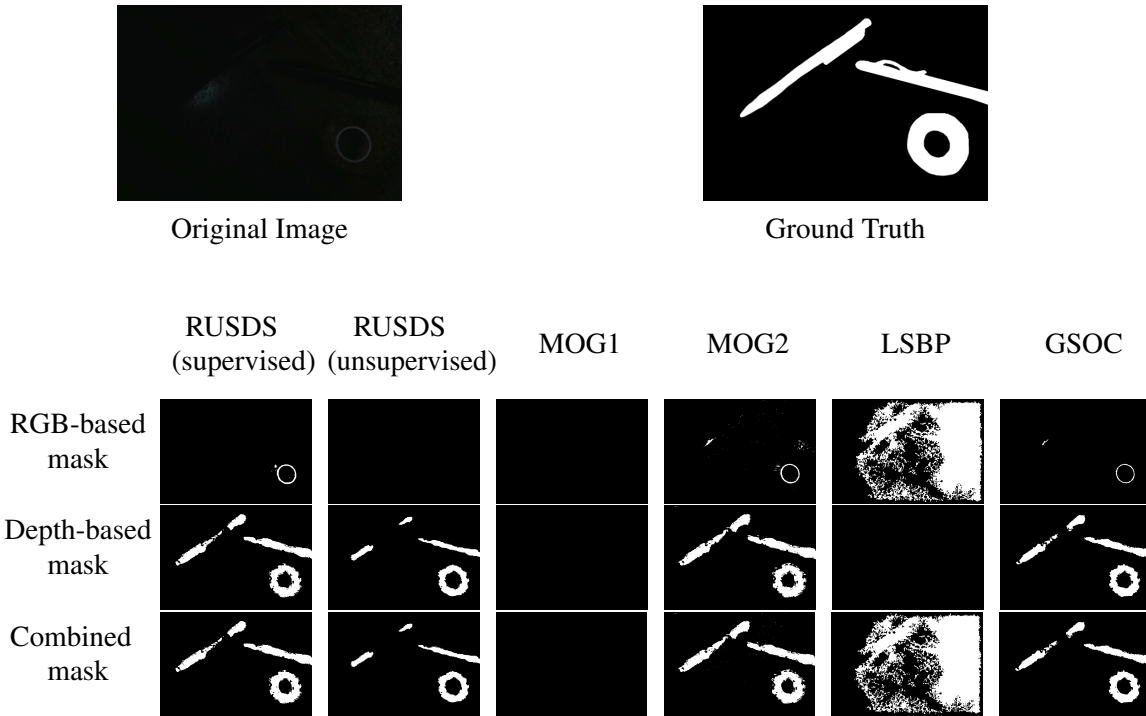


TABLE 4.8. The conditions of environment 3.2 can be thought of as an extension of environment 3.1, because environment 3.2 is also exemplifying a hard RGB, easy depth segmentation. The objects are segmented poorly when the background subtractors are exposed to only RGB data (row 1). This is because the objects make little to no RGB differences in the dark room. However, once the background subtractors are exposed to depth data in row 2, almost all of them are able to improve upon their initial RGB only segmentation. Curiously, MOG1 and LSBP failed this segmentation.

In environment 4, both RGB and depth-based masks are prone to false negatives. This is because environment 4 represents low RGB and low depth difference. This environment can be created by introducing camouflaged, highly 2D objects into the dark scene. Therefore, the objects/surface chosen for this environment were flat shards of paper on top of a same-colored sheet of paper. Depth masks are created in one of two ways: RUSDS uses depth subtraction (Section 3.2.2), while the other four background subtractors consider depth by performing RBSUDCI (Section 3.1).

**Environment 4:** Hard RGB and depth segmentation





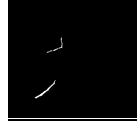
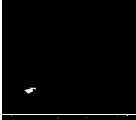
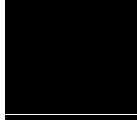

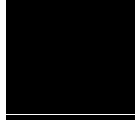



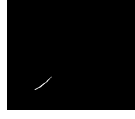

						
	Original Image	Ground Truth				
	RUSDS (supervised)	RUSDS (unsupervised)	MOG1	MOG2	LSBP	GSOC
RGB-based mask						
Depth-based mask						
Combined mask						

TABLE 4.10. The objects in environment 4 are segmented poorly, no matter what data the background subtractors are exposed to (rows 1,2, and 3). This is because the paper makes little to no RGB or depth differences in the scene. Curiously, MOG2 is able to detect some slight differences, while MOG1 is unable to detect any at all.

## 4.2 Quantitative: IOU performance comparison

The Intersection over Union, or IOU, is a metric for scoring how close a segmentation is to the ground truth segmentation. The IOU was calculated by taking the size of white pixel intersection with the ground truth, and then dividing it by the size of the white pixel union with the ground truth. In Table 4.11, each background subtraction algorithm was scored by calculating the IOU of the segmentation with respect to the ground truth segmentation. Higher scores indicate better segmentations.

Algorithm Name	Env 1	Env 2	Env 3.1	Env 3.2	Env 4	Average
RUSDS supervised	0.82	<b>0.94</b>	0.87	0.64	<b>0.13</b>	<b>0.68</b>
MOG2 combined	<b>0.83</b>	0.9	0.64	<b>0.67</b>	0.06	0.62
GSOC combined	<b>0.83</b>	0.89	0.66	0.62	0.01	0.602
MOG2 depth (RBSUDCI)	0.82	0.02	0.86	0.67	0.04	0.482
Simple depth subtraction supervised	0.81	0	0.87	0.63	0.01	0.464
GSOC depth (RBSUDCI)	<b>0.83</b>	0	0.87	0.61	0	0.462
RUSDS unsupervised	<b>0.83</b>	0.23	0.7	0.49	0	0.45
Simple depth subtraction unsupervised	<b>0.83</b>	0	0.87	0.49	0	0.438
MOG2 color	0.66	0.9	0.21	0.06	0.03	0.372
MOG1 combined	0.75	0.38	0.71	0	0	0.368
LSBP combined	0.5	0.38	0.71	0.23	0	0.364
GSOC color	0.77	0.89	0.06	0.03	0.01	0.352
RGB union supervised	0.65	0.94	0	0.05	0.12	0.352
MOG1 depth (RBSUDCI)	0.74	0	0.87	0	0	0.322
LSBP depth (RBSUDCI)	0.38	0	<b>0.88</b>	0	0	0.252
LSBP color	0.24	0.38	0	0.23	0	0.17
MOG1 color	0.15	0.38	0	0	0	0.106
RGB union unsupervised	0.19	0.23	0	0	0	0.084

TABLE 4.11. IOU scores of segmentation methods used

These results can be interpreted on a per-environment basis:

- Environment 1: Easy RGB and depth segmentation - Many methods performed excellently, all with an IOU score of 0.83. Every single one of these methods included some form of depth input.
- Environment 2: Easy RGB, hard depth segmentation - Supervised RUSDS had the slight edge over other methods due to the fact that the user was able to manually tune color and depth difference thresholds to produce the most optimal RUSDS-based segmentation. No other background subtraction methods were given the luxury of supervision. Not far behind, the next best methods are MOG2 and GSOC color. These two methods were the prime contributors to the success of the MOG2 and GSOC combined methods, since hardly no depth differences were present in this environment.

- Environment 3.1: Hard RGB, easy depth segmentation - The top performers of this environment were the depth-only methods. All of the RGB-based methods had trouble dealing with shadows. There was also little difference between the cube and surface colors.
- Environment 3.2: Dark room - Similar to environments 1 and 3, the top performers of this environment were the methods which considered depth. This is because depth differences are not dependent on the illumination of an environment, whilst RGB differences are.
- Environment 4: Hard RGB and depth segmentation - Every single method performed poorly in this environment. In hindsight, this is a virtually impossible segmentation to accomplish, because little to no RGB or depth differences are being created.

There are three methods that were significantly more versatile to the five environments presented, which is described by the average IOU column values on the right-hand side of the table above. These top three methods, ranked, are supervised RUSDS, MOG2 combined, then GSOC combined. It is crucial to understand that the MOG2 combined method is most likely a more versatile algorithm than supervised RUSDS, because MOG2 combined is a fully unsupervised method. Because of this, it is highly recommended that the MOG2 combined method is implemented in any most given systems over the RUSDS method. However, since the supervised RUSDS method was able to create highly accurate segmentations, the RUSDS method may prove useful within supervised conditions.

Finally, there was no combined method which performed worse than its RGB-only counterpart. This suggests that if there is a depth sensor present in the computer vision system, it is always advantageous to perform RBSUDCI as opposed to performing background subtraction on only the RGB image.

## CHAPTER 5

### Conclusion

---

This thesis has presented two novel methods for segmenting objects with RGB-D data - RGB Union + Simple Depth Subtraction (RUSDS), and RGB-based Background Subtraction Using Depth Colormap Input (RBSUDCI). These methods have been proven to be quantitatively robust to camouflaged objects, due to the fact that both methods consider depth differences, and camouflaged objects usually create substantial depth difference within a scene.

In conclusion, if there is a depth sensor present in a background-subtraction based computer vision system, it is always advantageous to perform RBSUDCI as opposed to performing background subtraction on only the RGB image. This strategy allows computer systems to better segment camouflaged objects before performing any post-processing of the background subtraction. Consequentially, computer vision systems using RBSUDCI will be able to detect camouflaged objects in real-time more accurately than systems which do not incorporate depth subtraction strategies. Furthermore, an unsupervised RUSDS is too unreliable to be worth implementing in real-world applications. Many more reliable unsupervised background subtraction algorithms already exist.

### 5.1 Future outlook

There are far more sophisticated ways to be considering depth differences. For example, an object of interest is very likely present if there is both a depth difference and a color difference within the image. However, it is less likely that an object of interest is present if there is only an RGB difference, and little to no depth difference. The same can be said for images with high depth differences, but very little RGB difference. This change in probability suggests that the method of simply taking the union of the RGB and depth segmentation may be sub-optimal when compared to a model which considers the ratios of RGB difference and depth difference on a per-pixel basis. Such adaptive methods were attempted, but were never successfully implemented in this research.

Finally, on a different note, further insights on RBSUDCI's performance can be made by evaluating it on a larger scale using a standardized benchmark.

## Bibliography

- [1] A.K. Jain. *Fundamentals of Digital Image Processing*. Pearson, 1988.
- [2] T.T.Mirnalinee P.Arjun. “Machine parts recognition and defect detection in automated assembly systems using computer vision techniques”. In: (2016).
- [3] Trembl et al. “Speeding up Semantic Segmentation for Autonomous Driving”. In: (2016).
- [4] D. Ciresan et al. “Multi-column deep neural network for traffic sign classification”. In: (2012).
- [5] V.Krger T.B Moeslund A. Hilton. “A survey of advances in vision-based human motion capture and analysis”. In: *Computer Vision and Image Understanding* 104 (2006), pp. 90–126.
- [6] W. Hu et al. “A survey on visual surveillance of object motion and behaviors”. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions* 34 (2004), pp. 334–352.
- [7] M.S. Lew et al. “Content-based multimedia information retrieval: State of the art and challenges”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 2 (2004), pp. 1–19.
- [8] Qiang Guo Xu. “Background Subtraction using Local SVD Binary Pattern”. In: (2016).
- [9] S. Fels D. Parks. “Evaluation of background subtraction algorithms with post-processing”. In: *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance* (2008), pp. 192–199.
- [10] D. Conte et al. “An Algorithm for Recovering Camouflage Errors on Moving People”. In: (2010).
- [11] R. Bowden P. KaewTraKulPong. “An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection”. In: (2001).
- [12] Z. Zivkovic. “Improved Adaptive Gaussian Mixture Model for Background Subtraction”. In: (2004).
- [13] Van der Heijden Z. Zivkovic. “Efficient adaptive density estimation per image pixel for the task of background subtraction”. In: (2005).
- [14] Goyette et al. *CDNET benchmark results*. 2012.
- [15] Goyette et al. *CDNET benchmark results*. 2014.
- [16] B. Panlal K. Sreedhar. “Enhancement of Images Using Morphological Transformations”. In: *International Journal of Computer Science Information Technology* (2012).