Human Subject Segmentation in Images with Complex Background

Esmaeil POURJAM

March 2016

Abstract

Human body segmentation has many applications in a wide variety of image processing tasks from intelligent vehicles to entertainment. A substantial amount of research has been done in the field of segmentation and it is still one of the active research areas, resulting in introduction of many innovative methods in literature. Still, until today a method that can automatically segment human subjects in different kinds of situations and with good accuracy, has not been introduced yet.

For a useful segmentation system to be realized, the following several problems that are thought to have more importance and effect 1) Shape variations due to human body movements, 2)Shape variations due to human wearing different clothes, 3) Variation in color and texture of the clothing, 4) Complexity of the scene.

The articulation problem can be considered as the most important problem. Because, the human body is made of multiple links and joints which leads to many variations in the general shape of the body during different movements in various kinds of situations. It is said that for completely modeling the human body a model with at least 20 degrees of freedom (20DOF) is necessary. This leads to a very sophisticated model which even the creation would prove to be very difficult. As a result, different approaches are usually selected instead of explicitly modeling the body ranging from 3D generative multi joint models to simple multi joint stick-man models. Also ,combination between the movements and change in the shape caused by human subjects wearing different types of clothes which can affect their general shape (rain coats, coats, T-shirts, normal shirts, and so on) adds to the complexity of the model.

Aside from this it can be said that the humans are the only species that can have numerous changes in their outer appearance by wearing different kinds of clothing. There numerous choices for humans in the clothes they use and even a single human changes his/her clothes in different situation and times. As a result a vast combination of colors and texture is created which again makes the task of modeling more problematic and sophisticated.

Complexity of the scene that a human subject is being recorded in is also one of important problems for a practical segmentation system. Even the simplest real-world cases for human eyes has proved to be quite challenging for the computer to understand and differentiate. This means that finding a model for the background will also become difficult. Especially if we use a moving camera the problem will ascend to another level of difficulty since now we need a dynamic model to cope with the situation.

Many of the recent methods try to use the graph-cut framework to solve the segmentation problem. Although powerful, these methods usually rely on a distance penalty term (intensity difference or RGB color distance). This term does not always lead to a good separation between two regions. For example, if two regions are close in color, even if they belong to two different objects, they will be grouped together, which is undesirable. Also, if one object has different parts with different colors, e.g. humans wearing clothes which have different colors and patterns, different parts will be segmented separately. Although this can be overcome by multiple inputs from user, the inherent problem would not be solved.

This thesis will try to address the above mentioned problems to propose automatic segmentation algorithms that can cope with these situations. For overcoming the changes in the shape and color we propose a method using the conventional Statistical Shape Models (SSMs) and Grab-cut segmentation algorithm. SSM uses the statistical framework to model changes in the shape of an object in Eigen space and by analyzing it and finding the most dominant changes it can even try to generate some new shapes with the properties of the original one, while the Grab-cut segmentation algorithm tries to learn image color distributions and segment the foreground object based on that in an Markov random fields frame work. By connecting these two methods through a feedback system it becomes possible to propose a coarse to fine scheme for model generation and refinement which in combination with Grab-cut leads to accurate segmentation results for human subjects.

To cast aside the color/texture problem we propose a system which makes use of a human probability map, super-pixels and Grab-cut framework. The main idea comes from jigsaw puzzle game. If we divide the image into regions based on their color/texture, each part of the human body then, becomes like a piece of puzzle. So we can think of an image as a puzzle with multiple pieces in which the human body occupies some of them. By selecting the right pieces, we can have a somewhat rough (or even fine) shape of the body and by using the Grab-cut, we can segment the human subject accurately. Following this idea, we show that not only the system becomes automatic but also the accuracy of the system is improved. We also show that, just by using the information of a single image, it is possible to achieve segmentation results with accuracy comparable to the state-of-the-art and much better than traditional methods while having relatively simpler model. It is also good to note that this method can be used in both automatic and interactive segmentation manner.

In the name of Allah the compassionate, the merciful

Acknowledgements

This dissertation is formally submitted for fulfilling partial requirements for the degree of Doctor of Science in Engineering from Nagoya University. To me it also serves as a reminder of 5 best years of my life in Japan from 2011 to 2016. Actually comming to Japan was full of new experinces for me as I had to live outside of my home contry for a relatively long period of time and for the first time. During these years I came across many poeple, some had greate influence on my life. I am really gratefull to each and every one of them.

I want to express my sincierest gratitude to my suprvisor Prof. Dr. Hiroshi Murase. Not only he accepted me to his laboratory, but also with his continious help, support, encoragement, and kindness thrughout my 5 years of reserach, made it possible for me to finish my PhD. and also have great life experince in Japan. I could not have hoped for a better advisor.

I am really gratefull to Associate Prof. Dr. Ichiro Ide who always was of great help in the process of proofreading of my works, especially this thesis and with his invaluble comments made the work to be the best possible quality. I am still amazed on how fast, accurate and useful his comments and corections were.

Associate Prof. Dr. Daisuke Deguchi always provided me with invaluable assistance whenever I was troubled with the technical aspects of my research and have put me in the right direction. His insightful comments during my research saved me from lots of trouble. I would like to sincerely thank him for all his helps during my stay in the Murase Laboratory.

I also am really grateful to Prof. Dr. Kensaku Mori for his great comments and help in proofreading of this thesis. He kindly provided me his invaluable comments on the thesis several times especially when he was very busy with his work.

Special thanks also go to the other members of Murase Laboratory: the secretaries, Mrs. Fumiyo Kaba and Mrs. Hiromi Tanaka, for processing almost all of my Japanese documents and for making my life run smoothly in Japan. I also thank Prof. Dr. Yoshito Mekada, Associate Prof. Dr. Tomokazu Takahashi, Dr. Yasutomo Kawanishi, Dr.

Keisuke Doman, Dr. Haruya Kyutoku, for their helps, and other students for their interesting discussions and helps during my study at Nagoya University.

I also want to thank everyone in Education Center for International Students of Nagoya University who with their help and assistance during the Japanese classes, made me accustomed with the interesting Japanese language.

I would also like to acknowledge the support from the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) in the form of Monbukagakusho scholarship throughout my years of study at Nagoya University as Research Student (2011–2012) and PhD. student (2012–2015).

I also want to dedicate this thesis to My wife and daughter whom their support and encouragement was my main motivation to move forward during these years, and to my father and mother who made every effort in their power for me since birth and made me the person I am today.

Finally I thank Allah who with his guidance and help all this has become possible.

March 2016,
Nagoya

Contents

List of Figures

List of Tables

Abbreviations

AAM	**A**ctive **A**ppearance **M**odel
ASM	**A**ctive **S**hape **M**odel
CRF	**C**onditional **R**andom **F**ield
DAG	**D**irected **A**cyclic **G**raph
DDMCMC	**D**ata **D**riven **M**arkov **C**hain **M**onte **C**arlo
EM	**E**xpectation **M**aximization
GAC	**G**eometric **A**ctive **C**ontour
GC	**G**rab-**C**ut
GMM	**G**aussian **M**ixture **M**odel
GVF	**G**radient **V**ector **F**low
HOG	**H**istogram of **O**riented **G**radients
LBP	**L**ocal **B**inary **P**attern
MAP	**M**aximum **A** **P**osteriori
MRF	**M**arkov **R**andom **F**ield
PAC	**P**arametric **A**ctive **C**ontour
PDF	**P**robability **D**istribution **F**unction
SOM	**S**elf **O**rganizing **M**ap
SSFSeg	**S**tatistical **S**hape model **F**eedback **Seg**mentation
SSM	**S**tatistical **S**hape **M**odel
SVM	**S**upport **V**ector **M**achine

Chapter 1

Introduction

1.1 Overview

Object segmentation is one the most important tasks in image processing and vision with a generous amount of research being done in the field. Image segmentation itself is defined as “the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze” [1]. This definition shows how wide-spread can be the applications of image segmentation as it is mentioned by Agarwal et al. [2] as, “Segmentation can be considered the first step and key issue in object recognition, scene understanding and image understanding”. In simpler terms, the main goal of object segmentation methods can be explained as the capability of separating (segmenting) the desired object(s) from the input image. From different types of objects existing in the natural environment, humans have caught attention of many researchers. By advances in the technology and appearance of intelligent systems, the need for human-machine interaction has also became necessary. Making the systems recognize the presence of human, understand the human command, and be able to interact with humans, have proved to be of vital necessity. Many kinds of applications can be imagined for the mentioned capabilities of which, some has been realized nowadays. These applications can vary from very important rescue missions in disasters like earthquakes, driving assistance systems (some of companies have started rolling out cars with automatic navigation system) or intelligent houses to entertainment systems like Microsoft's Kinect. Still, humans are considered to be an articulated type of object meaning there would be many changes in their body appearance and shape as they move around or when they are pictured from different aspects. Furthermore, humans wear various types of clothing in different occasions and situations which add to the complication. The mentioned problems alongside common ones in object segmentation like illumination changes (day/afternoon/night, under sunlight/in shadow) as shown in Figure 1.1, registration noise/blur in Figure 1.2, complexity of the scene in Figure 1.3, and so on, will lead to a problem which has challenged many researchers for a long time. Still, a method that can accurately segment a human subject, adapt to different situations, and work automatically, has not yet been proposed. It is also good to note that despite the rich literature in the field of object segmentation, to the knowledge of the author, there is no survey that covers the methods available for human subject segmentation. This might be because human body segmentation is usually considered as a start point or preprocessing step for other vision tasks like face detection/recognition, action recognition, pose/gesture estimation, and so on.

As a result, here, a brief literature review on different methods and algorithms available for image/object segmentation is provided below followed by introduction to some works done on human segmentation.

1.2 Image/Object Segmentation

As mentioned before, literature on image/object segmentation is so rich that some researchers have tried to make surveys to introduce available methods in the field. There are surveys like those by Agarwal et al. [2], Vantaram et al. [4], Cheng et al. [5], Sridevi et al. [6], Freixenet et al. [7], and many others in which a comprehensive review and explanation is done. There are even works like the one by Haralik et al. [8] and Prantl et al. [9] that try to explain the terminology commonly used in literature.

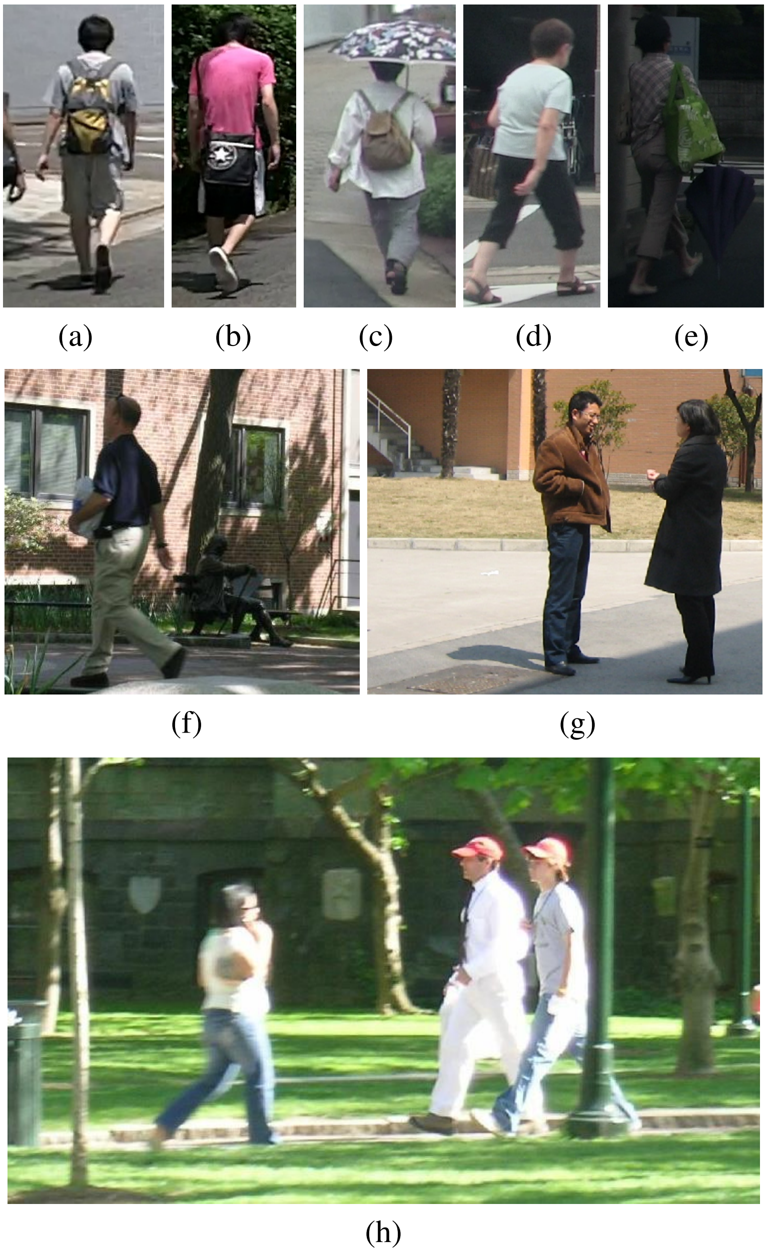


FIGURE 1.1: Effect of illumination changes during different times of the day. (a)&(b) Normal day light, (c)&(d) Cloudy day, (e) Rainy day, (f) Shadow cast, (g) Two subjects in the same place. One of the subjects is in a normal situation while the other one is under shadow. (h) Image taken in a bright day. Images (a) ~ (e) are from the author's private dataset while images (f) ~ (h) are from PennFudan dataset [3].

Categorization of methods can be done in different ways based on the opinions of authors and intended applications. One general approach for example, is to categorize methods based on the type of the image they work on (grayscale vs. color), Using single-scale representation or multi-scale one, need for user interaction (Automatic vs. Interactive) and so on. This usually leads to a few major categories in which different

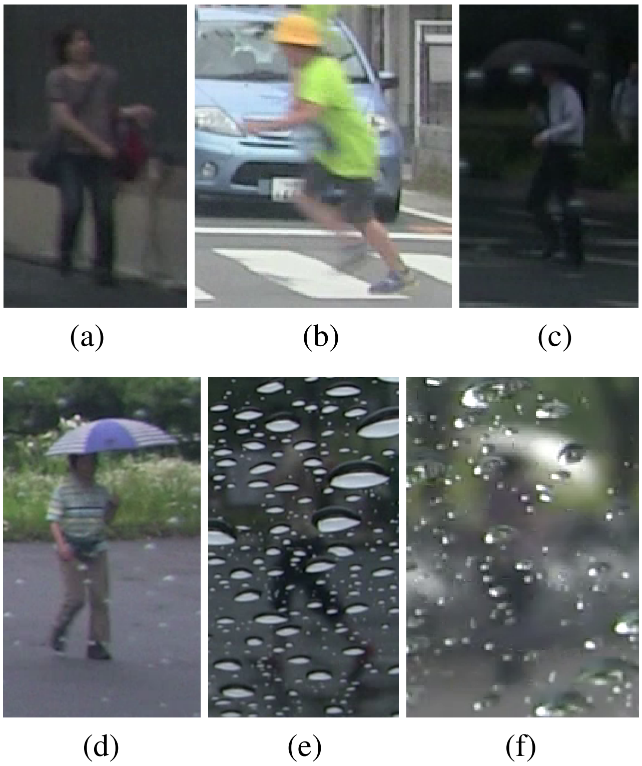
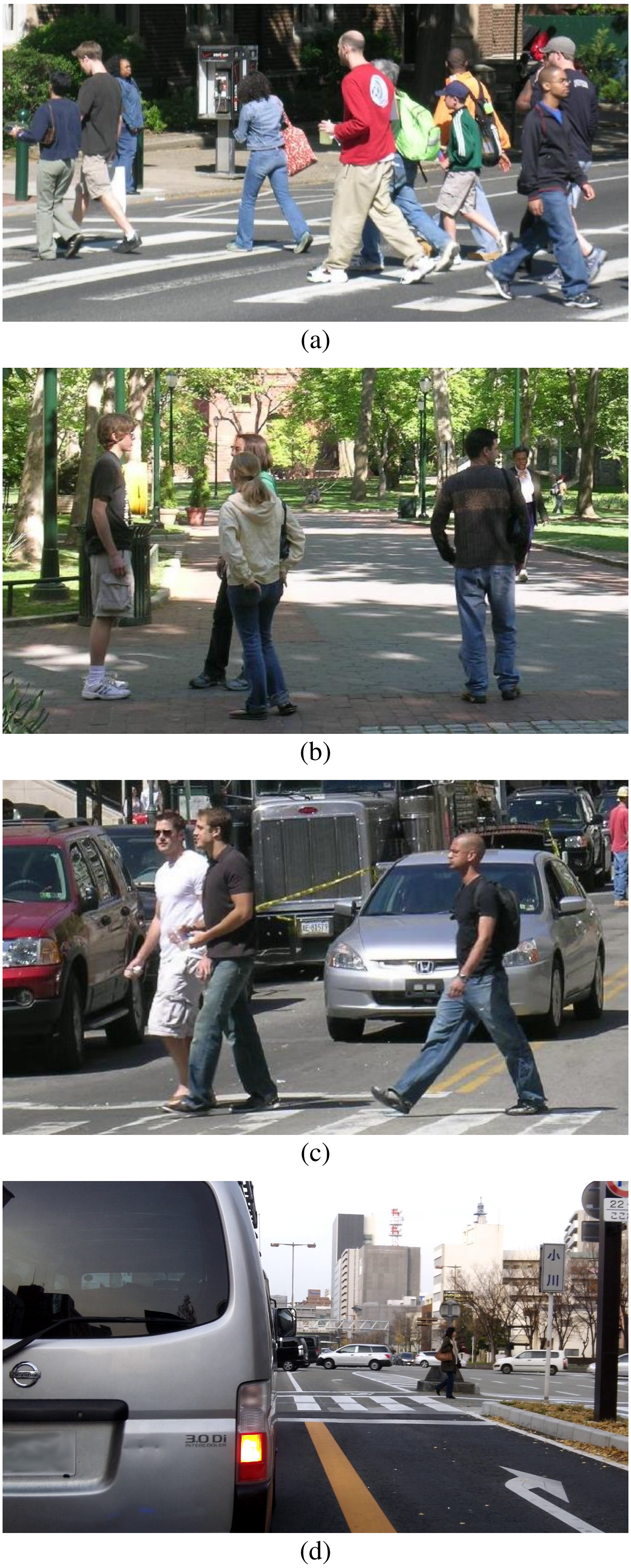


Figure 1.2: Different kinds of noise in images, (a) Blur due to camera movement, (b) Blur due to fast object movement, (c)&(d) Noise due to rain with little effect on the subject, (e)&(f) Heavy noise which makes the object unidentifiable. Images from author's private dataset.

methods are presented without any specification about the procedure they use to perform the segmentation. For example, all of methods can be associated to be either automatic (unsupervised) [10–17] or interactive [18–28] segmentation. The former tries to find and segment the object-of-interest automatically without any interference and usually needs initialization around the object-of-interest, while the latter needs user interaction in different levels of segmentation process to avoid miss-segmentations. The automatic segmentation algorithms have the advantage of being free from external interference (user interaction), but the main problem with them is the initialization and low segmentation accuracy in most of the cases. On the other hand, although interactive methods provide relatively accurate segmentations, user input is necessary for achieving satisfactory results which renders them useless for automatic applications.



(a)

(b)

(c)

(d)

FIGURE 1.3: Complex scenes containing human subjects. (a) ~ (c) are from PennFudan dataset [3] while (d) is from author's private dataset.

Automatic segmentation algorithms are useful for cases that the numbers of images/-subjects to be segmented are not predefined (e.g. driver assistance systems, entertainment systems) or the number is too big to be segmented manually (e.g. image/video archives), while interactive algorithms are preferred for applications in which the number of images or subjects to be segmented are limited so that the user interaction cost would be feasible. In these methods, usually, user interaction is tried to be reduced as much as possible.

Vantram et al. [4] have made a detailed survey on image segmentation methods. They have proposed their categorization in an abstracted chart partially presented in Figure 1.4. They have divided the segmentation methods into three main categories.

- Spatially blind: Methods that do not make use of spatial information about image pixels and try to perform the segmentation based on some kind of feature.
- Spatially guided: Methods that make use of spatial relation between image pixels and try to segment an image into homogeneous regions/groups with respect to space.
- Miscellaneous methods: Other methods that cannot explicitly be categorized in either of the above.

1.2.1 Spatially Blind Techniques

These methods try to make use of some kind of feature for segmentation. They usually belong to either of the following two sub-categories mentioned here:

- **Clustering:** Methods which assume the digital image as a point cloud in 1D (grayscale image) or 3D (color image) and then partition the cloud into clusters of meaningful data based on a specific criterion. In case of color image, different kinds of color spaces such as RGB, $L^*u^*v^*$, YC_bC_r, HSI (HSV), LAB, YIQ, . . . might be used. The result should be so that similar pixels be in one cluster,

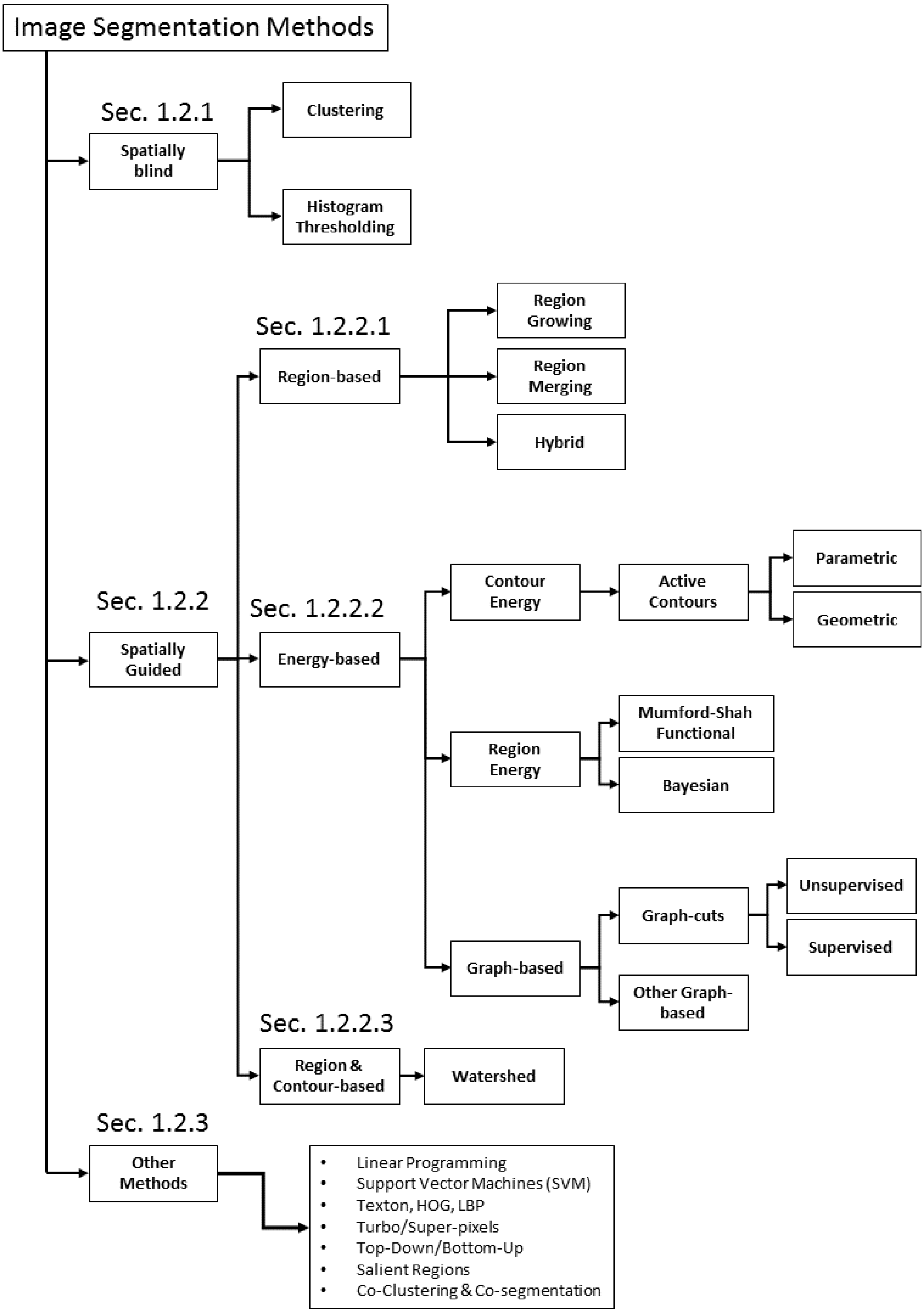


FIGURE 1.4: Low-level segmentation approaches [4].

while each cluster would be distinguishable from the others. Some famous methods like K-means clustering [29], Mean shift clustering [30, 31], Self-Organizing Maps (SOM) [32], or Fuzzy C-means [33] fall under this sub-category.

- **Histogram thresholding:** Methods that try to segment an image based on the information obtained from the histogram of that image. By analyzing the shape of the peaks and valleys, it is possible to segment image into different parts. Because of its simple nature, the methods in this sub-category have become popular for gray-scale image segmentation. Still, using them for color images has proved to be difficult, since thresholding in a 3D space is required. Methods like those by Tan et al. [34] on histogram thresholding or Bhattacharyya et al. [35] on adaptive thresholding fall under this sub-category.

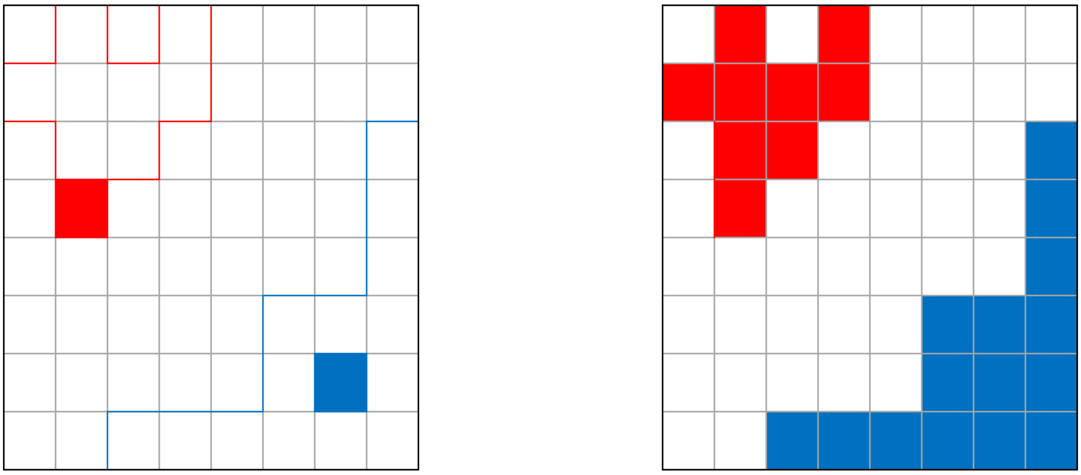
1.2.2 Spatially Guided Techniques

These methods try to make groups of pixels with spacial homogeneity, despite any kind of characteristics pixels might have in the feature space. They are usually divided into the following three sub-categories:

1.2.2.1 Region-Based Methods

Methods using region-based information usually apply one of "region growing", "region merging", "region splitting" methods, or a combination of them for the segmentation.

- **Region growing:** The process starts by selecting one or a group of pixels as seeds and then this group is expanded using some kind of homogeneity criterion, iteratively. Methods like [36–40] try to utilize this algorithm for segmentation. Figure 1.5 shows an example of how region growing works.
- **Region splitting:** The process usually starts with an inhomogeneous segmentation of the image and tries to find homogeneous groups by splitting the regions



(a) Some seeds are selected in the image

(b) The result of region growing after some iterations

Figure 1.5: Example of region growing in an image.

repeatedly. Note that like "Region growing", the homogeneity criterion is defined beforehand. Various methods have been proposed in this sub-category from them [41, 42] can be mentioned.

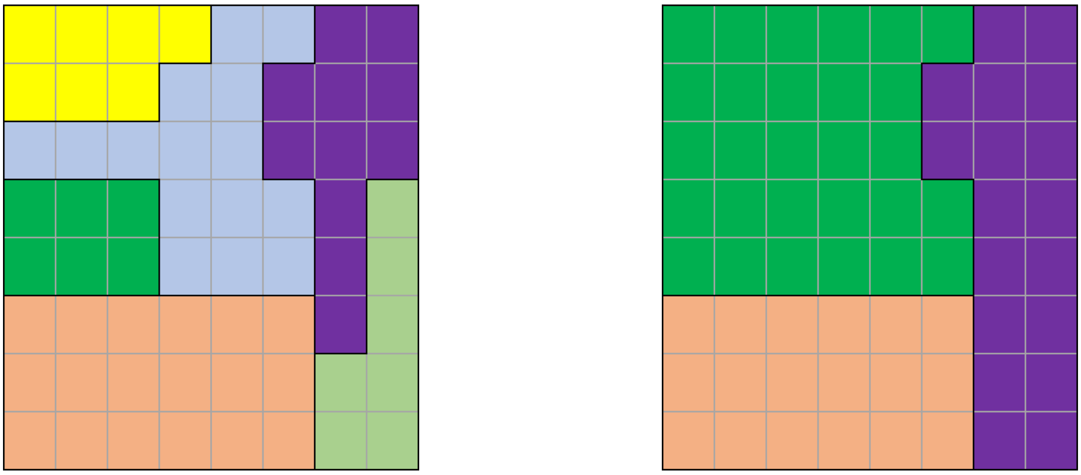
- **Region merging:** In contrast to "Region splitting", here, the given regions are merged to find a more meaningful result based on some criterion, as depicted in Figure 1.6. Works like [43–45] present some methods that apply region merging for image segmentation.

It is also good to note that applying region growing/splitting will lead to over-segmentation so usually, a scenario based on split/merge is devised to prevent the problem.

1.2.2.2 Energy-Based Methods

Many of the famous segmentation methods fall in this category. From them we can note the following.

- **Contour energy minimizers:** These methods try to perform the segmentation task using an energy minimization scheme based on an energy model defined to



(a) Initial regions in the image

(b) The result of region merging after some iterations

FIGURE 1.6: Example of region merging in an image.

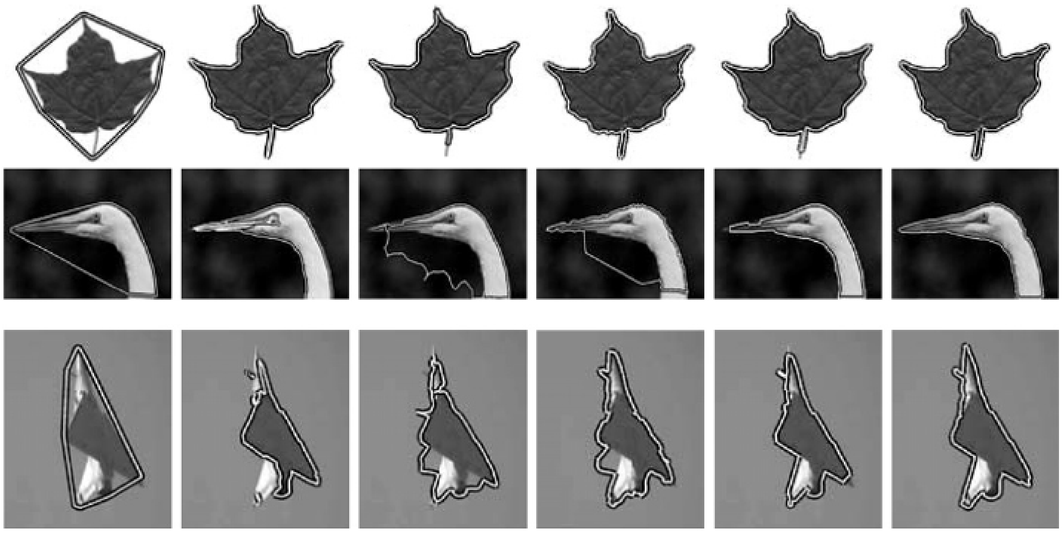


FIGURE 1.7: Examples of image segmentation using Active contours. (a) User initialization, (b) Chan-Vese [10] (c) GVF [11], (d) Harris-based GVF [46], (e) Vector Field Convolution (VFC) [46], (f) Harris-based VFC (HVFC) [46]. Image taken from [46].

have minimum on the boundary of the object. Active contour (snake) [15] is the most famous algorithm in this category which try to segment the object-of-interest by minimizing a defined contour energy model. They have two major types, "Parametric Active Contours" (PACs) like the original active contour by Kass et al. [15], Gradient Vector Flow (GVF) active contour [11, 47], Balloon snakes [48] and many others like [46, 49–51], and "Geometric Active Contours" (GACs) like the methods proposed in [10, 12, 13, 52–55]. Figure 1.7 shows examples of how different methods perform segmentation.

- **Region energy minimizers:** These methods perform the segmentation by minimizing energy model defined to have minimum values based on some regional features like the methods using “Mumford-Shah” functional [10, 56–59] or the ones using “Bayesian” models [60–63]. Mumford-Shah functional has relieved the curve evolution methods (GAC and PAC) from restriction of using edge information as a stop criterion. The methods using “Bayesian” models usually try to model regional features as Markov Random Fields (MRFs) and use the Maximum A Posterior (MAP) method used in that framework for the energy minimization procedure. From the methods using Mumford-Shah functional, Level Sets [10] are very famous while from the ones using Bayesian framework, Conditional Random Fields (CRFs) [64] are widely known. These methods use statistical inference and prior information about the image and are usually used when image has some non-deterministic features, like textures, statistical noise, etc., which is one of shortcomings of the conventional methods.

- **Graph-based energy minimizers:** These methods have become very popular specially when they are used for interactive segmentation, due to their segmentation accuracy. Usually the image is considered to be an undirected graph (explained in 2.1). Each image pixel will be then become a node of the graph with some edges connecting it to its neighbors. Also, each edge will be assigned a weight which shows the degree of similarity between two nodes based on some feature. The goal will then be to segment the graph to subgraphs which are considered meaningful with respect to a defined criteria. Many of recent famous methods like Normalized-cut [65], Graph-cut [18], Grab-cut [19], One-cut [66] or those in [14, 16, 20–24, 67–69] will fall under this sub-category. More details will be presented in Chapter 2.

1.2.2.3 Region-Contour Based Methods

The most famous method here is the Watershed [70, 71] which tries to segment an image into some “catchment basins” based on region and contour information available in the

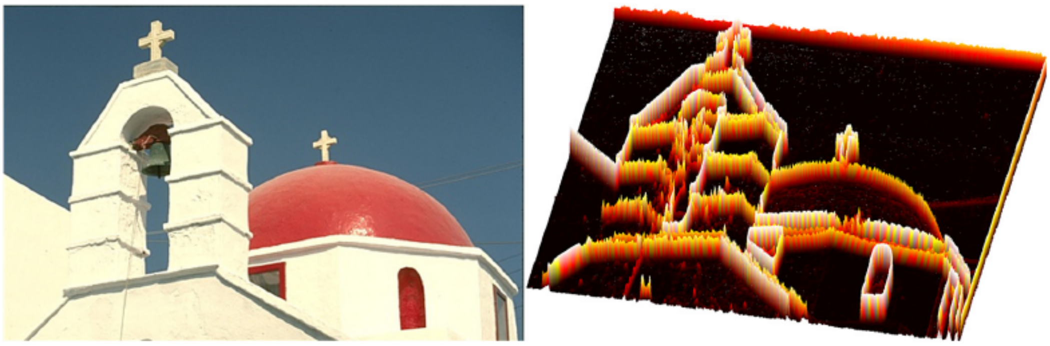


FIGURE 1.8: Example of the topographic relief map used by Watershed (Image taken from [4]).

image by viewing it as a topographic relief map (like the one in Figure 1.8).

In these maps, the third dimension (the first and the second are x and y coordinates respectively) would be the feature (usually intensity, gradient, and so on). Each pixel is assumed to be stationed in this terrain and then moves towards the local minima like when water goes down a valley. The pixels that go down the same region form the mentioned catchment basins that are called “watersheds”. Watershed has a simple mathematical algorithm which makes it easy to implement and optimize which leads to a very fast segmentation process. However, it tends to over-segment and there is need for further process to achieve the desired results. On the contrary, because of its fast segmentation process, it is possible to use Watershed as a preprocessing stage for different methods at low cost. Many methods like marker-based Watershed [72], texture-gradient Watershed [73], Watersnakes [74], work of Vanhamel et al. [75], and multi-resolution Watershed [76], have been proposed which take this approach alongside different algorithms, trying to segment images more effectively.

1.2.3 Other Methods

Aside from the mentioned methods, there are some that cannot be classified as either of categories mentioned before. Still, some of them are well known or have some interesting concepts and are noteworthy. In the survey by Vantaram et al. [4], they are categorized as “miscellaneous” segmentation methods. From those types of works, we can note:

- Fuzzy-based methods such as the ones involving fuzzy homogeneity [77, 78] and fuzzy region completion [79] processes.

- Supervised methods using adaptive weighted distances [80], spline regression [81], geodesic matting [82], and linear programming [83].

- Methods using specialized image features; namely, quaternions [84, 85], textons [86, 87], Histogram of Oriented Gradients (HOG) [88], and Local Binary Patterns (LBP) [89, 90].

- Methods that use turbo-pixel or super-pixel based representations of an image [91, 92].

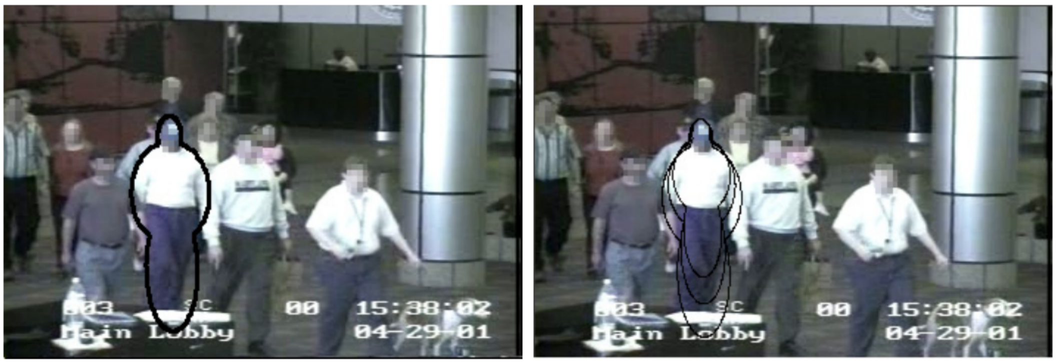
- Methods that consider the segmentation as a classification and perform the task using sophisticated classifiers such as Support Vector Machines (SVMs) [93, 94], or employ specialized properties of images [95].

- Methods that use statistical principles [96], information bottleneck method [97], and algorithms that consider segmentation as a task of finding perceptually salient groupings like [98].

- Methods that use co-clustering strategies, which combine multiple segmentations into one improved result [99] as well as co-segmentation methods that jointly segment multiple images, which contain a common object [100].

1.3 Human Subject Segmentation

Now lets take a brief look at some methods that use human subject segmentation as a part of their tasks. There are some surveys like [101] that explain methods used for action recognition and segmentation, or the survey of Gandhi et al. [29] which focuses on the pedestrian detection systems in which some use segmentation for the sake of detection or tracking.



(a) (b)

FIGURE 1.9: Human body model using three ellipses. (a) Model is composed of three parts: head (top circle), torso (middle ellipse), and legs (downside ellipse). (b) Matching the model to actual human body by detecting the head and adjusting the model to match the whole body (Image taken from [104]).

Popa et al. [102] propose a method for human body segmentation using structural shape priors. For this, they make use of bottom-up region detection constraints and on-the-fly shape prior construction using recently available Human3.6M [103] to propose "Constraint Parametric Problem Dependent Cuts with Shape Matching, Alignment and Fusion (CPDC-MAF)". Using this, they try to overcome the problems of multi-view and partial-view of the human subject segmentation. For segmentation, they first use a human region detector to achieve a set of human region candidates. After that they use a very large dataset and a matching, alignment fusion technique to create the shape prior based on the acquired candidates. At last, they use the estimated prior to segment the human subject. The main problem here, is the dependency of the system on the candidates to be found which are not always correct and also need for a very large training set (100,000 sets of silhouettes, skeletal structures and pose information) for creating the shape prior based on the candidate regions.

Mori et al. [105] use body part model segmentation for recovering the body pose. Their idea is to find candidates for salient parts of human body (torso, upper and lower parts of legs, and hands, in their work) in the input image, and try to find a reasonable combination of these parts to segment the human subject. For this, they use a boundary object detector proposed by Martin et al. [106] and Normalized-cuts [65], and divide the image into some regions and super-pixels. After that, they try to find limbs by using contour, shape, shading and focus cues. They also search for the torso using same cues except shading. Then, they use some constraint on the relative widths and lengths of limbs, and symmetry between clothing in each part to connect the limbs to the torso. Using the generated model, they try to segment the human subject and obtain its pose. This system focuses on finding the salient limbs and torso candidates to extend them into full body parts and connect them together to achieve the body pose. Since here, normalized-cut is used with pre-defined number of segments (40 for salient parts and 200 for full details) some details might be lost in the process which might not be considered useful for pose estimation but might be useful for other tasks like action recognition. Also, some parts like head, hands, and feet are not considered in the pose model which not only affects the accurate pose estimation, but also when these parts are segmented by chance, the proposed method cannot cope with them and sometimes makes wrong decisions.

Sharma et al. [107] propose a method for simultaneously detecting and segmenting a human subject by integrating both bottom-up and top-down frameworks for segmentation. In the top-down stage, a rough estimation of the subject is acquired using the contour information of an input image and a Probability Distribution Function (PDF) learned from training data to find human related information. In the bottom-up stage, interaction between extracted contour features like smooth continuity, closer, and so on, are considered. Finally the found information is included into a Markov Random Field (MRF) and by using min-cut algorithm, the foreground is estimated. This method tries to make the segmentation by just incorporating the edge and boundary information of a human body, which in some cases are not very clear and easy to achieve. In such cases, their method will most probably fail to segment and localize the subject.

Lin et al. [108] detect and segment human subjects by proposing a hierarchical part-template matching algorithm in combination with discriminative learning to create a generic human detector. It uses both a global template based detector and a local part-based detector to improve the accuracy. In their proposed method, the global templates are divided into a local part-based tree like the one presented in Figure 1.10, and the observations are matched to this tree for finding suitable candidates as human parts which then results in the final segmentation. Although the body parts model used here, is more complex than that of the previous methods, generating the part-base tree and

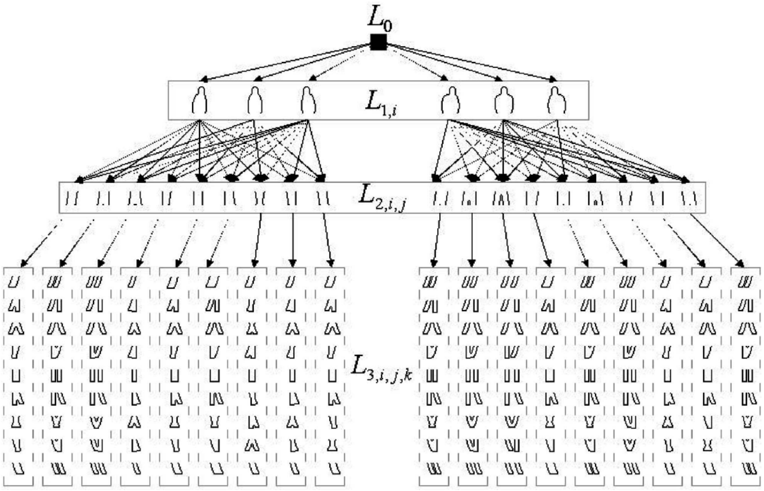


FIGURE 1.10: The part template tree used in Lin et al. (Image taken from [108]).

matching the parts to the actual shape limits on how much detail can be fed into the tree. As mentioned by the authors, although adding more details to the tree is possible, the matching cost increases linearly with the numbers of temples existing in the tree while adding more details just slightly improves the performance of the system. Also all of the input images are downscaled to the size of 64 × 128 pixels. Matching and segmenting in this size might prove to be accurate but this does not mean that the segmentation accuracy will be the same on the subject in actual size.

1.4 Thesis contribution

In this chapter, some existing methods in the field of object and human segmentation has been reviewed. As mentioned before, the literature on this subject is so rich that mentioning all of available methods is out of the scope of this thesis, so, some major categories based on the work of Vantaram et al. [4] have been introduced. As a general categorization, segmentation algorithms can be devised based on different aspects of the image they use (color/gray, single/multi-scale, single/multi-attribute, and so on), need for supervision or their principal of operation. A brief introduction to each one has been presented in this chapter.

As mentioned before, human body segmentation has many kinds of applications and at the same time has several problems to be solved. In many of the real-world applications, usually a large amount of data has to be processed. As examples of this, we can mention applications like video archiving, creating train or test datasets for other vision tasks, or more importantly, in driver assistance or automatic vehicle navigation systems which nowadays has become very famous. In the first example, a fair amount of segmentation accuracy would suffice while the amount of time needed for the process is usually not of the main concern (while it has to be reasonable), while in the second one, the accuracy can play a crucial role in the assessment of the methods that the dataset is going to be used for and usually, the more the data we have the more accurate the system will get. Again, the time consumption is not the main concern here. In contrast to the other two, the driver assistance or automatic navigation systems need both high accuracy and real-time performance to be of practical use. A system with high accuracy, reliability and real-time performance can considerably affect the problems related to accidents thus improving the driver safety.

In Section 1.2 various methods have been introduced that perform the segmentation task with different methodologies. As mentioned there, Object segmentation methods can be generally divided into 3 sub-categories of 1) Spatially blind, 2) Spatially guided and, 3) Other methods. From them spatially blind techniques are most commonly used for basic operations and are not fit for the complex scenarios like the one in this thesis. The last sub-category can cover different methods and techniques and making a general statement about it would be difficult. Still, in case of human subject segmentation, they won't be very useful unless they are used in combination with other methods as usually they usually incorporate a single aspect of the object to-be-segmented. In contrast, energy-based methods in spatially guided subcategory have proved to be useful in complex scenarios. Table 1.1 presents a very brief comparison of methods available in spatially guided subcategory.

As mentioned before there are different problems in vision field which one have to overcome when performing the segmentation task. From them, illumination changes (day/afternoon/night, under sunlight/in shadow) as shown in Figure 1.1, registration noise/blur in Figure 1.2, complexity of the scene in Figure 1.3 can be mentioned. For human subject segmentation, problems like changes in body appearance and shape due

TABLE 1.1: Comparison between different spatially guided object segmentation methods. + means the feature is supported, −/+ means it is possible to support the feature with additional implementation, and − means the feature is not supported.

	Region-based			Energy-based					
Methods	Region growing	Region merging	Hybrid	Active contours	Mumford-Shah	Baysian	Graph-cut	Other graph-based	Watershed
Automatic initialization	−	−	−	−/+	−/+	−/+	−/+	−/+	+
Various prior information inclusion	−/+	−/+	−/+	−/+	−/+	+	+	+	−/+
Complex object	−/+	−/+	−/+	−	−	−	+	+	+
Computation time	+	+	+	−	−/+	−/+	+	−/+	+
Robust against noise	−	−	−	−/+	−/+	+	+	+	−
Robust against weak boundaries	−	−	−	−/+	−/+	+	+	+	−
Multiple object segmentation	−/+	−/+	−/+	−	+	+	+	+	−/+

to movement, changes in clothing color/texture and appearance, Changes in appearance due to being viewed from different angles, will also be added to the mentioned ones.

Since considering all of the available problems would make the task very difficult, in this thesis, the following problems that are thought to have more importance and effect, will be considered.

1. Shape variations due to human body movements
2. Shape variations due to human wearing different clothes
3. Variation in color and texture of the clothing
4. Complexity of the scene

The first problem relates to many variations in the shape of a human body, especially when the subject moves or the video is recorded from different views and angles. This makes the modeling task difficult and a model that can cope with all of the variations can be very complex.

The second problem comes from the fact that humans use various types of clothes in different situations. Combination between the movements and change in the shape caused by human subjects wearing different types of clothes which can affect their general shape (rain coats, coats, T-shirts, normal shirts, and so on) adds to the complexity of the model.

Another problem about clothing is their color and texture. An unimaginable combination of clothes is available in the society. These combinations not only change between different people but even for one person, they will change in different occasions. Clothes are also different between genders, so for example, a model designed specifically for men might not work for women.

The next problem which makes the task more sophisticated is the complexity of the scene that a human subject is being recorded in. Even the simplest real-world cases for human eyes has proved to be quite challenging for the computer to understand and differentiate. This means that finding a model for the background will also become difficult. Especially if we use a moving camera the problem will ascend to another level of difficulty since now we need a dynamic model to cope with the situation.

The most practical way to overcome these problems, which is also used in this thesis, is to incorporate some kind of information about the human subject to the segmentation process. As indicated in the Table 1.1, Graph-cut based methods have better capability of incorporating different kinds of prior information and are better suited for complex object segmentation. Most of the methods that use Graph-cut framework (the Graph-cut included) use interaction as their initialization. From those, Grab-cut by [19] has made the user interaction as simple as inputing a polygon around the object-of-interest and used an iterative energy minimization algorithm. it also utilizes the RGB color information in contrast to the intensity information used in original Graph-cut by Boykov et al. [18]. It is also possible to use the Grab-cut in automatic segmentation by making a predefined map of labels for the input image which is the reason that it has been uses as a basis method in the proposed methods in this thesis.

The main goal of this thesis is to propose methods for automatic human subject segmentation. Different applications can be imagined for a this kind of methods some can be mentioned as follows

- Virtual reality systems (also entertainment systems) - Generating more realistic simulations.
- Driver assistance systems - Detecting the pedestrians and their intention and assisting the driver for safe driving.
- Automatic navigations systems - The same as above but for accident prevention and so on.
- Video archiving
- Surveillance - Detecting intruders.

To achieve this goal and with the aim to solve the above problems (except for the color/texture problem) in Chapter 3, a method is proposed which utilizes Statistical Shape Models and Grab-cut framework for automatic segmentation. By taking a coarse-to-fine model refinement through feedback between the shape model generation and segmentation stages, the accuracy of the system is proved to be much higher than the original Grab-cut method. Also, the need for user interaction is cast aside by the utilization of new shape generation and use of tri-maps in the segmentation stage. The model refinement can be a time consuming procedure, and also the initialization of generated models is not an easy task and their matching to the actual human shape might not be always possible. Also, using just color feature for subject segmentation sometimes is not sufficient as there are cases in which the color between the foreground object and its background is very similar. In case of human body, there are also times that because of the variations in the texture of clothes or sudden changes between the color of clothes and skin will lead to miss-segmentations (parts of clothing or body become the background). As a result, in Chapter 4, the problem is solved by a different approach through using human body probability map, super-pixels, texture feature, and Grab-cut framework. By using the texture information, it is possible to overcome the problem of

Grab-cut which just uses color similarity as the main segmentation factor and further improve the accuracy of the system. Here, the main goal is to turn the input image into a puzzle and then find the parts that have human parts inside using the probability map provided beforehand. This way, there will not be any need for model initialization and matching. Also, since there is no need for multi-stage model refinement, the time consumption of the system is drastically decreased compared to the one in Chapter 3.

The rest of this thesis is arranged as follows:

- Chapter 2 will be an introduction to Markov Random Fields (MRF), which with their versatility and capability for contextual information encoding has proved to be an effective tool in the field of vision and image processing, graph theory which is the basis for many recent methods in image segmentation and Grab-cut, the method that utilizes these in one framework for interactive image segmentation. Grab-cut is also the basis for methods proposed in this thesis.

- Chapter 3 gives a detailed explanation about the proposed automatic human subject segmentation algorithm which utilizes a slightly modified version of Grab-cut in combination with Statistical Shape Models (SSM) for more accurate segmentation.

- Chapter 4 presents the details of another automatic segmentation system for humans which utilizes a human probability map, super pixels, and Grab-cut to increase the segmentation accuracy.

- Chapter 5 is the last chapter of the thesis and concludes the research done in this thesis.

Chapter 2

Introduction to the Technologies Used in Proposed Methods

In this chapter, some background information about Grab-cut and Graph-cut frameworks will be presented. For understanding them, some background knowledge is needed about graph theory, Markov random fields and min-cut/max-flow graph partitioning method. As a result, some brief explanation will be presented about them and then explanation about Graph-cut and Grab-cut will be presented.

2.1 Graph Theory Related Terms

In image processing, usually an image is considered as an analogue mapping of the real 3D world into a 2D plane, and thus digital images become the discrete sampling of this mapping. In this regard, the digital images are composed of some pixels which are “finite” in their number and usually have a “rectangular” formation. These properties have made it possible to consider the digital images as some plane grids in which it will be possible to define graphs and use information of each pixel and its surrounding as a tool to perform different image processing tasks. From this point onward, the word is used “image” for digital images as nowadays most of image processing tasks are performed on digital images and analogue images are now considered a thing of the past.

In all graph-based image segmentation methods, an image is represented by its equal "graph". In simple terms, a graph, noted by $\mathcal{G}$, is a finite set of some "vertices" ("nodes") noted as $\mathcal{V}$, and some connections between the nodes called "edges" or "arcs" noted as $\mathcal{E}$. It might also be called an "incident" to the vertice it is connected to. By the mentioned notations, a graph will be presented as follows.

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \tag{2.1}$$

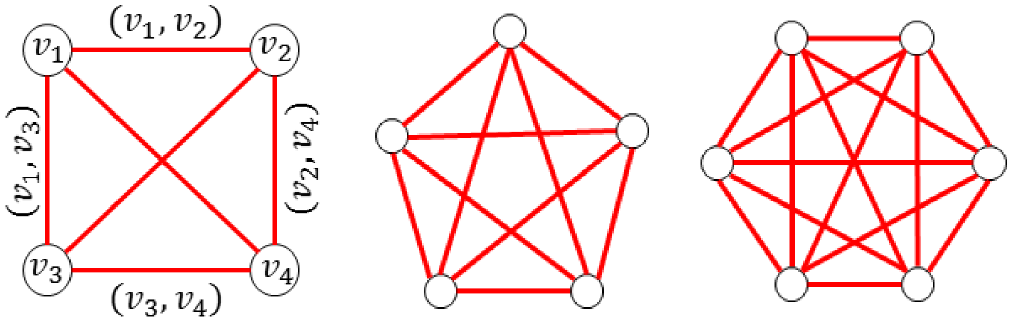
$$\mathcal{V} = \{v_1, \ldots, v_m\} \tag{2.2}$$

$$(v_i, v_j) \in \mathcal{E} \tag{2.3}$$

where m is the number of nodes in the graph and (v_i, v_j) denotes the edge connecting vertices v_i and v_j together. $\mathcal{E}$ contains the set of links connecting the vertices together like the example depicted in Figure 2.1 (a). The "order" of the graph $|\mathcal{V}|$ would be described as the number of its vertices, while the "size" of the graph $|\mathcal{E}|$ would be the number of the edges. A graph would be called "complete" ("fully connected") of order n (noted as K_n) if all n vertices are "adjacent" to each other. Two vertices are said to be adjacent if there is an edge connecting them together, so in a complete graph of order n, all of nodes are connected to each other by means of edges, thus the size of the graph would become $|\mathcal{E}| = \frac{n\times(n-1)}{2}$ as shown in Figure 2.1. A graph is also called "planar" if it is possible to draw it on a plane, given no edges intersect each other.

A "neighborhood" is a set of adjacent vertices defined around a certain vertice v_i. It is said to be "open" if v_i is not included (usually "neighborhood" refers to this type of set) and "close" if v_i is also included in the set. Please note that here, we are talking about simple graphs, so there is just one edge between two vertices and there is no "loop" (an edge that its start and end points are one vertice) or the like in the graph.

A "clique" is a subset of vertices in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ which form a complete subgraph (each two vertices are adjacent). It is called "maximal" if there is no possibility of adding another vertice to the subset and it would become "maximum" if there is not



(a) K_4 with $|\mathcal{E}| = 6$ and $|\mathcal{V}| = 4$

(b) K_5 with $|\mathcal{E}| = 10$ and $|\mathcal{V}| = 5$

(c) K_6 with $|\mathcal{E}| = 15$ and $|\mathcal{V}| = 6$

FIGURE 2.1: Example of complete graphs where each node is connected to all other nodes.

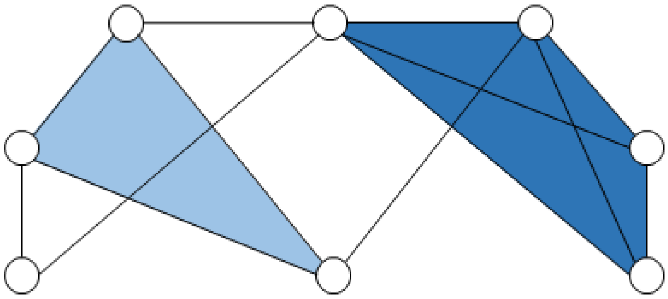
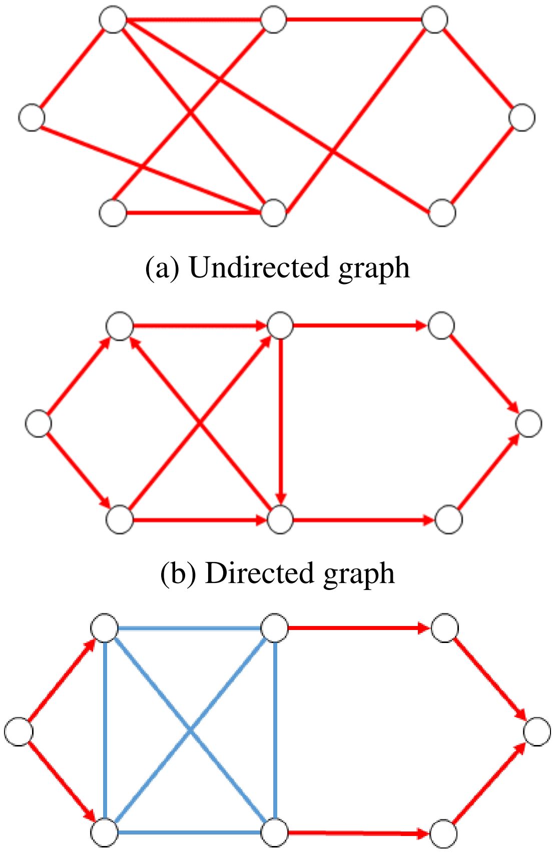


FIGURE 2.2: A graph with 8 × 1-vertex cliques (the vertices), 13 × 2-vertex cliques (the edges), 5 × 3-vertex cliques (the light and dark blue triangles), and 1 × 4-vertex clique (just the dark blue area). The four light and dark blue triangles form the maximal cliques. The dark blue 4-vertex clique is both maximum and maximal, and the clique number of the graph is 4.

another clique in the graph that has more vertices than this one. The number of vertices in the maximum clique is called the "clique number" noted by ω. Figure 2.2 shows an example of how cliques are enumerated in a graph.

Graphs can also be "undirected" or "directed". In the latter, direction of movement between the vertices is important and the edges are usually represented as arrows starting from an "initial vertice" ("tail") and ending to a "terminal vertice" ("head"), while in the former, direction is irrelevant. There are also "mixed" graphs containing both mentioned types (Figure 2.3).

Another term which is common in graph theory is "weighted" graph in which a weight w_{ij} is assigned to each edge (somewhat like the edge capacity in the flow network). The weight is usually a real number, but depending on the application, some



(a) Undirected graph

(b) Directed graph

(c) Mixed graph

Figure 2.3: Example of different kinds of graphs.

constraints might apply, e.g. in Dijkstra's algorithm [109], the weights must be positive. The "weight of path" or "weight of tree" is then calculated as the sum of weights of the selected edges on the graph. In Graph-cut based methods, the constructed graph is always a weighted graph.

"Cutting" a graph means partitioning it into two disjoint subgraphs. Each cut creates a set of edges that has an end point in each partition (the set is called a "cut-set"). In the flow network mentioned before, an "s-t cut" becomes a cut that puts the source and the sink nodes in different partitions. The cut-set here will only have the edges going from the source to the sink. The capacity of the cut is defined to be the sum of the capacity of edges in the cut-set, while its "size" (also called weight of cut-set some times) will be the number of edges in the cut-set. A "minimum cut" is a cut with the smallest possible size, while the "maximum cut" is a cut with the biggest possible size in the graph as depicted in Figure 2.4.

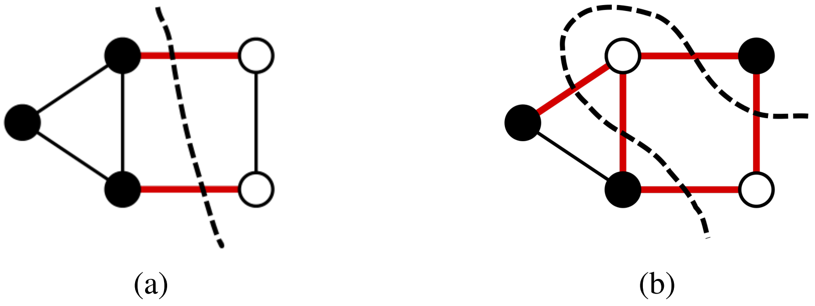


Figure 2.4: Example of minimum and maximum cut in an undirected unweighted graph. (a)Minimum cut with size=2, (b) maximum cut of same graph with size=5

.

2.2 Markov Random Fields and Graph-based Segmentation

In the field of computer vision, there are many tasks like image restoration, stereo disparity calculation, surface reconstruction, texture analysis, optical flow, shape from X, or segmentation that can be posed and modeled as labeling or energy minimization problems. Usually, the energy model for these kinds of problems is assumed to be represented as the following:

$$E(\mathcal{M}) = E_{\text{data}}(\mathcal{M}) + E_{\text{smoothness}}(\mathcal{M}) \tag{2.4}$$

which has a data term $E_{\text{data}}(\mathcal{M})$ for adapting the model $\mathcal{M}$ to the observed input data, and a smoothness term $E_{\text{smoothness}}(\mathcal{M})$ which is defined based on our knowledge about the solutions.

For tasks like image segmentation, contextual information about the object to be segmented can play a vital role in the accuracy and the correctness of the result. Thus, using such kinds of information like texture and object features can be a very useful asset. The main problem here would be on how to model these kinds of information in a way that makes us able to use them for example, in an energy minimization framework.

Markov Random Fields (MRFs) are one of the most suitable thus famous methods for encoding contextual information into a mathematical framework. As described in the work by Ki et al. [110], the following four reasons are the main motives to use MRFs for solving the problem of the tasks mentioned at the beginning of this section:

- For various problems, using defined principals in a systematic manner instead of some heuristic modeling will become possible.
- It will be possible to evaluate the results of the system in a quantitative manner instead of a qualitative one.
- Incorporating various types of contextual information will become possible.
- MRF-based methods tend to be local, which makes them ideal for hardware parallelization.

An MRF framework is usually used in conjunction with statistical estimation theories to formulate the needed objective function as a probabilistic model. For this purpose, Bayesian framework is one of the most famous ones used. In this framework, the optimal solution is defined in the form of “Maximum A Posteriori” (MAP) estimates to obtain the best parameters from random observations. For this, usually a joint distribution should be derived which is a difficult task, but thanks to the Hammersly-Clifford theorem which indicates that the joint distribution of an MRF can be equivalently described as a Gibbs distribution, the problem takes a rather simpler form. Thus, five steps for MRF modeling in accordance to Li et al. [110] will become:

1. Consider the vision task as a labeling problem in which a label configuration would be the solution.
2. Pose it as a Bayesian labeling problem where optimal solution would be in a MAP labeling configuration.
3. Use Gibbs distribution to model joint priori distributions.
4. Using the assumed observation of data, find the likelihood density.
5. Use the Bayesian rule to calculate the posterior distribution of labeling configuration.
6. Calculate the cost of obtained labeling configuration.

2.2.1 Markov Random Field (MRF)

An MRF is usually defined by the following terms:

- A set of sites

  $$\mathcal{S} = \{1, \ldots, m\} \tag{2.5}$$

  with $1, \ldots, m$ as indices representing a point or region in Euclidean space (a pixel or a feature like object corner, contour, and so on). An image of size $n \times n$ can be represented in form of sites as

  $$\mathcal{S} = \{(i, j) | 1 \leq i, j \leq n\} \tag{2.6}$$

  Since models are usually unordered, the definition in Equation 2.6 will look like Equation 2.5 when we index each pixel with a number and with $m = n \times n$. Sites can be categorized as "regular" like image pixels or "irregular" like image features (edges, corners and so on) extracted from image and relation between them is defined by their "neighborhood".

- A set of random variables (sometimes called weights) with a variable assigned to each site $\mathcal{W} = \{w_1, \ldots, w_m\}$.

- A set of neighbors in each site $\mathcal{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$. Each set defines the relation between the site and the neighboring sites.

The model must adhere to interdependency requirement of Markov property which is:

$$\forall i \leq m; P(w_i | w_{\mathcal{S} \setminus i}) = P(w_i | w_{\mathcal{N}_i}) \tag{2.7}$$

in which "$\mathcal{S} \setminus i$" means the full set of sites except site i.

This means that the model is conditionally independent of all other variables given its neighbors (the same with conditional independence in an undirected graph). As a result, an MRF can be considered as an undirected graph so that:

$$P(\mathcal{W}) = \frac{1}{Z}\prod_{i=1}^{J}\phi[\mathcal{W}_{C_j}], \tag{2.8}$$

where $\phi[\bullet]$ is a potential function which returns a non-negative value. This value is decided by the state of variables in a clique $C_j \subset 1, \ldots, m$. Z is a normalizing constant and is used to make sure the results abide to the probability rules. Rewriting the equation 2.8 in Gibbs format we will have:

$$P(\mathcal{W}) = \frac{1}{Z}\exp\Big[\sum_{j=1}^{J}\log\left[\phi[\mathcal{W}_{C_j}]\right]\Big] \tag{2.9}$$

$$\tag{2.10}$$

In above equation $\psi = -\log\left[\phi[\bullet]\right]$ is a cost function with a real value.

2.2.2 MAP-MRF Labeling

Many of the image processing task can be defined like segmentation; edge detection, corner detection, etc. can be posed as a labeling problem in which a label must be assigned to each site (pixel or feature). In a labeling problem, we have a set of labels which can be continuous like

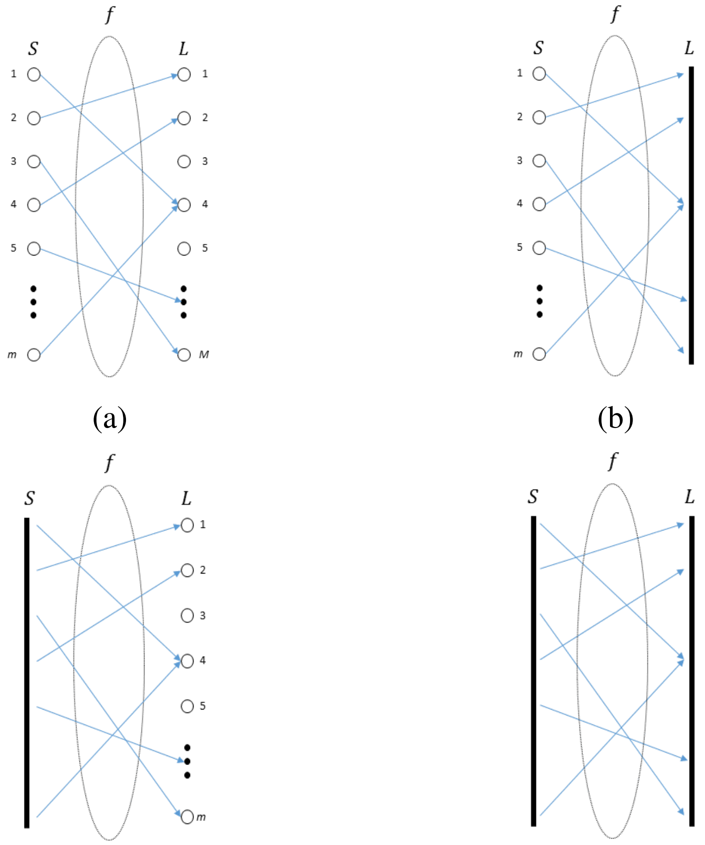
$$\mathcal{L} = [X_1, X_2] \subset \mathbb{R} \tag{2.11}$$

in which X_1 and X_2 show the start and ending points of the range like analogue pixel intensity. Also labels can be discrete like

$$\mathcal{L} = \{l_1, \ldots, l_M\} \tag{2.12}$$

or in a more compact form

$$\mathcal{L} = \{1, \ldots, M\} \tag{2.13}$$



(a) (b)

(d) (e)

Figure 2.5: Four types of possible combinations for the set of sites $\mathcal{S}$ and set of labels $\mathcal{L}$. (a) both $\mathcal{S}$ and $\mathcal{L}$ are discrete, (b) $\mathcal{S}$ is discrete and $\mathcal{L}$ is continuous, (c) $\mathcal{S}$ is continuous and $\mathcal{L}$ is discrete, (d) both $\mathcal{S}$ and $\mathcal{L}$ are continuous. In all of cases, the labeling set f can be considered as a mapping from set of sites $\mathcal{S}$ to set of labels $\mathcal{L}$.

Usually a label is defined as an event that can happen to a site. For example, in case of image segmentation, the set of labels will become, $\mathcal{L} = \{\text{foreground}, \text{background}\}$. In the labeling problem, a set like

$$f = \{f_1, \ldots, f_m\} \tag{2.14}$$

is called a "labeling" on $\mathcal{S}$ in terms of labels in $\mathcal{L}$. It also can be considered a mapping from $\mathcal{S}$ to $\mathcal{L}$ as shown in Figure 2.5 (a).

As mentioned before, Bayesian rules are used here to extract the posterior distribution. For this, we have to minimize the difference between the estimated labeling (f^*)

and the original solution (f) which is expressed in the form of a risk function like

$$R(f^*) = \int_{f \in \mathbb{F}} C(f^*, f) P(f|x) df \tag{2.15}$$

in which x is the observation, $C(f^*, f)$ is a cost function and $P(f|x)$ is the posterior distribution. For achieving $P(f|x)$, we can use Bayes rule which indicates

$$P(f|x) = \frac{p(x|f)P(f)}{p(d)} \tag{2.16}$$

where $P(f)$ is the posterior probability of f, $p(d|f)$ is the probability distribution of the observed data x (likelihood of f given x) and $p(d)$ is the density of x. As for the cost function, we can set it like

$$C(f^*, f) = \begin{cases} 0 & \text{if } ||f^* - f|| \leq \delta \\ 1 & \text{otherwise} \end{cases} \tag{2.17}$$

The Bayes risk function will then be

$$R(f^*) = \int_{||f^* - f|| > \delta} P(f|d) df = 1 - \int_{||f^* - f|| \leq \delta} P(f|d) df \tag{2.18}$$

and if $\delta \to 0$, it can be approximated as

$$R(f^*) = 1 - \kappa P(f|d) \tag{2.19}$$

where κ is a volume of space with all labellings $||f^* - f|| \leq \delta$ inside. Minimizing this is equivalent to the maximization of posterior probability which leads to

$$f^* = \arg \max_{f \in \mathbb{F}} \{P(f|d)\} \tag{2.20}$$

Since $p(d)$ is constant (observation is given), $P(f|d)$ becomes relative to joint distribution. As a result, the MAP estimation can be written as

$$f^* = \arg \max_{f \in \mathbb{F}} p(d|f) P(f) \tag{2.21}$$

Here, for the sake of explanation, a common example in image processing is used; the noise removal problem. We want to reconstruct an image based on available noisy data. If we assume that the image surface is flat, the joint distribution can be written as

$$P(f) = \frac{1}{Z} \exp\left(- \sum_{i} \sum_{i' \in \{i-1, i+1\}} (f_i - f_{i'})^2 \right) \tag{2.22}$$

If we assume that the available noisy data is a combination of the values on the actual image surface plus an independent Gaussian noise distribution, we can write the observations as

$$x_i = f_i + e_i \tag{2.23}$$

$$e_i \sim N(\mu, \sigma^2) \tag{2.24}$$

The probability of distribution $P(d|f)$ will then become

$$P(d|f) = \frac{1}{\prod_{i=1}^{m} \sqrt{2\pi\sigma^2}} \exp\{-U(d|f)\} \tag{2.25}$$

$$U(d|f) = \sum_{i=1}^{m} \frac{(f_i - x_i)^2}{2\sigma^2} \tag{2.26}$$

where $U(d|f)$ is the “likelihood energy”. With this, the posterior probability becomes

$$P(f|d) \propto \exp[-U(f|d)] \tag{2.27}$$

$$U(f|d) = U(d|f) + U(f) \tag{2.28}$$

Thus MAP estimate can be calculated by using

$$f^* = \arg \min_{f} U(f|d) \tag{2.29}$$

Since in this example, we have only σ_i to determine, when decided, the MAP-MRF will be completely defined.

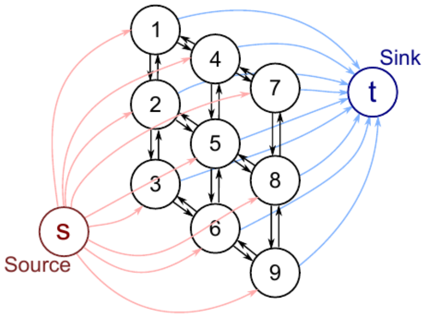


FIGURE 2.6: Structure of the graph created for MAP estimation. Each pixel of the 3×3 image is assigned to a node in the graph and directed edges connect nodes together. Aside from that, some edges connect the *source* node to all nodes in the graph and some edges connect nodes of the graph to the *sink* node. Image taken from [111].

2.2.3 Max-flow/Min-cut Algorithm

For inferring the MAP tasks, we usually try to define them as a “maximum flow” problem in the graphs so that the “minimum cut” (equivalent to maximum flow) of the graph corresponds to the MAP solution. For this, we map each pixel in the image as a node in a directed graph along with two additional nodes as “source” and “sink” which are connected to every node in the graph as depicted in Figure 2.6. Directions of these additional edges would be from the source towards the sink. Now, all we have to do is to assign weights to the edges of the created graph so that the cost of each cut on the graph would match the cost of labeling in MRF. The problem of cutting the graph then, can be solved by using maximum flow and minimum cut methods usually known as max-flow/min-cut algorithm. Assume that we have a directed graph like the one in Figure 2.7 with limited capacity for each edge and we want to transfer some quantity (“flow”) from the source node to the sink node. The goal in max-flow would be to put through as much flow as possible in the network while not overloading the edges of the graph. The solution is said to be found when each path from the source to the sink contains at least one saturated edge. Now if we remove the saturated edges from the graph, the path from the source to the sink will be completely lost. In another meaning, these edges create a set that can separate the source and the sink nodes which is also noted to as a “cut” on the graph (refer to section 2.1 for explanation). Please also note that, this set consists of edges that are saturated, meaning their cost (capacity) is the least among all other edges of the graph. As a result, it will be a cut with minimum

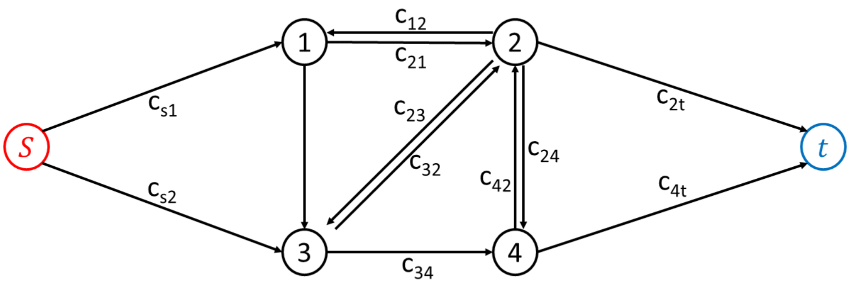


FIGURE 2.7: Example of a graph with 6 nodes aside source and sink vertices in which max-flow problem should be solved.

cost on the graph which is called "minimum cut" or "min-cut" for short. From this, it is understandable that the max-flow and the min-cut are equivalent and finding either of them will lead to the same result. For solving the max-flow problem, different methods exist, but a simple way would be "augmenting paths" as described in [111]. Briefly presenting, we first select a path in the graph from the source to the sink and push the flow into it until one of the edges is saturated. We remove the saturated edge and subtract its cost from all edges of the graph, and repeat the process until there is no path from the source to the sink. The total flow being transferred from the source to the sink would be the maximum possible flow, while the saturated edges create the minimum cut set. This process is depicted in more details in Figure 2.8. Finding a suitable cut of the graph is equivalent to finding the solution of the MAP problem which also leads to the desired result for our task (image denoising, segmentation, and so on).

2.3 Grab-cut Segmentation Framework

As briefly presented in 1.2.2.2, Grab-cut is an interactive graph-based image segmentation method introduced by Rother et al. [19]. It improves the Graph-cut method by Boykov et al. [18] by incorporating color information and iterative energy minimization framework which leads to better performance and wider area of applicability for the system. The user interaction is also relaxed to just selecting a rectangle around the object-of-interest as in Figure **??**.

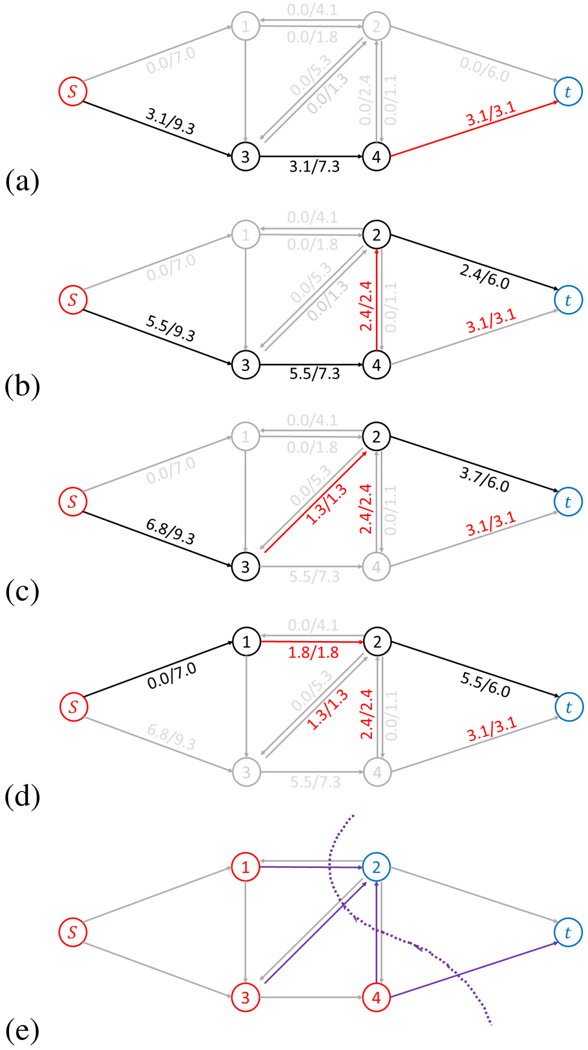


Figure 2.8: Example of performing max-flow/min-cut. (a) One of the available paths in the graph is selected and flow is pushed through it until one of edges of that path is saturated. (b) Next path with spare capacity is selected and the same thing is done, (c)&(d) Same process is done to all available paths in the graph and (e) A set of saturated edges that separate the source from the sink is selected as the solution to the min-cut problem.

Their method starts with defining the Gibbs energy as follows

$$E(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) + V(\underline{\alpha}, \mathbf{z}), \tag{2.30}$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ is the RGB color values of the image, $\underline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N)$ is an array of opacity values $0 \leq \alpha_n \leq 1$, but for hard segmentation it is either 1 (foreground) or 0 (background). $\underline{\theta}$ shows the foreground and background models expressed by Gaussian Mixture Models (GMM).

The task is then to find a set of labels $\underline{\alpha}$ using MAP-MRF. So, like Equation 2.29, we should find

$$\hat{\underline{\alpha}} = \arg \min_{\underline{\alpha}} \mathbf{E}(\underline{\alpha}, \underline{\theta}) \tag{2.31}$$

The color information is also modeled in the image as GMMs of full-covariance and with K=5 components. One GMM is used to model the foreground and one for the background. Each pixel in the image will be assigned to one of the GMM components belonging to either foreground or background. A vector $\mathbf{k} = \{k_1, \ldots, k_N\}, k_n \in \{1, \ldots, K\}$ is used to encode the mentioned assignment. The data term $D(\alpha_n, k_n, \underline{\theta}, z_n)$ is

$$D\left(\alpha_n, k_n, \underline{\theta}, z_n\right) = -\log p(z_n \mid \alpha_n, k_n, \underline{\theta}) - \log \pi(\alpha_n, k_n) \tag{2.32}$$

in which $p(\bullet)$ is the Gaussian probability distribution, Σ is the covariance matrix, and $\pi(\bullet)$ is the mixture weighting coefficients. Expanding the $p(\bullet)$ part of the above equation, we will have

$$\begin{aligned} D(\alpha_n, k_n, \underline{\theta}, z_n) = & -\log \pi(\alpha_n, k_n) + \frac{1}{2} \log \det\left(\Sigma(\alpha_n, k_n)\right) \\ & + \frac{1}{2}[z_n - \mu(\alpha_n, k_n)]^T \Sigma(\alpha_n, k_n)^{-1} [z_n - \mu(\alpha_n, k_n)] \end{aligned} \tag{2.33}$$

The model will also be like

$$\underline{\theta} = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k),\ \alpha = \{0, 1\},\ k = \{1, \ldots, K\}\} \tag{2.34}$$

The smoothness term is used in the same way as by Boykov et al. [18]. The only difference is that, here, the 3D color space is used and Euclidean distance between two pixels is calculated in RGB space instead of using the image intensity on grayscale image pixels.

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n)\in\mathbf{C}} \frac{\exp\{-\beta \|z_m - z_n\|^2\}}{\operatorname{dist}(m, n)} \tag{2.35}$$

The above equation is valid given $\alpha_n \neq \alpha_m$. **C** is the set of neighboring pixels and dist(•) is the Euclidean distance function. Parameter β controls the smoothness term in a way that the smoothing in high contrast areas be avoided. According to Boykov et al. [18], it should be set to

$$\beta = \frac{1}{\langle 2\|z_m - z_n\|^2 \rangle} \tag{2.36}$$

in which $\langle \bullet \rangle$ is expectation over an image sample and parameter γ is set to 50 based on some training done by Blake et al. [112].

Although Boykov et al. use min-cut algorithm only once for image segmentation, Rother et al. devised an iterative method which allows them to refine their model incrementally. The basic structure of their work is presented in Figure 2.9.

The steps presented in Figure 2.9 is straight forward. After a user puts the rectangle (or polygon) around the object-of-interest, the tri-map T is initialized setting $\alpha_n = 0$ for $n \in T_B$ and $\alpha_n = 1$ for $n \in T_U$. GMM components are also initialized for foreground and background using this initial labeling (usually background is learned from $\alpha = 0$ labels and foreground is leaned from $\alpha = 1$ labels. Using the famous k-means clustering algorithm, the $K = 5$ components in each GMM is specified based on color distributions). In the first step of iterative energy minimization, each pixel of the image will be assigned to one of the components in either the foreground or the background GMM. In Step 2, mean $\mu(\alpha, k)$ and covariance $\Sigma(\alpha, k)$ are calculated. The weights are set to be $\pi(\alpha, k) = |F(k)|/\sum_k |F(k)|$ where $|F(k)|$ indicates the size of the set. Step 3 will perform the usual max-flow/min-cut algorithm for estimating the α labels. Rother et al. [19] also mention that since each of the three steps can be considered as energy minimization on **E** with respect to **k**, $\underline{\alpha}$, and $\underline{\theta}$, the algorithm is guaranteed to converge.

2.3.1 Tri-maps & User Interaction

Since an iterative energy minimization scheme is used here, the user will gain more freedom for interacting in the process of choosing the labels of pixels by the system. In other meaning, the user will be able to provide the system with an incomplete tri-map

Initialization

- Foreground, background and unknown region tri-maps (T_F, T_B, T_U) are initialized from the user selected area. Outside of the area will be T_B while the inside becomes T_U and $T_F = \phi$.
- Set $\alpha_i = 0$ for $i \in T_B$ and $\alpha_i = 1$ for $i \in T_U$ (initial labeling).
- Initialize background and foreground GMMs from α.

Energy Minimization

1. Assign each pixel in T_U to a GMM component.

$$k_i = \arg\min_{k_i} D_i(\alpha_i, k_i, \theta, z_i)$$

2. Learn GMM parameters.

$$\underline{\theta} = \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$

3. Estimate segmentation using min-cut algorithm

$$\min_{\{\alpha_i : i \in T_U\}} \min_{\mathbf{k}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$

4. Repeat from step1 until convergence.

User corrections

- Correction: Change α_i to 0 or 1 (background & foreground), Update tri-map and perform segmentation estimation once.
- Refinement: Perform energy minimization process again.

FIGURE 2.9: Basic flow of the Grab-cut algorithm proposed by Rother et al. [19].

and the system tries to converge to the correct answer based on that incomplete input. For example, the user can just specify the background (T_B in Figure 2.9) pixels and the system will find the label for other pixels. This is the form which is well known when working with Grab-cut in which the user selects a rectangle around the object-of-interest. The same can be done for a foreground object by just selecting a part of the foreground (T_F) and letting the system find the rest. It should be noted that in both cases when a pixel is selected by the user as either foreground or background in the incomplete tri-map, they cannot be changed during the energy minimization process since the whole system is working based the information provided by them. If the segmentation result is not satisfactory, the user then will be able to select some seeds as foreground and background, and perform the minimization again. This process can be repeated until a satisfactory result is achieved.

2.4 Summary

In this chapter, the fundamentals of the methods used in this thesis was presented. At first some glossary definition on graph theory was reviewed to make the reader more familiar with the definition needed for understanding how the graph-based segmentation systems work. After that, a brief explanation about the Markov Random Fields (MRFs) which make the base for many problem solving algorithms for graph-based vision tasks was explained. Using MRFs makes it possible to convert the contextual information of the image and the object (edges, gradients, and so on) into a fine mathematical framework and incorporate them in the vision tasks leading to more effective solutions. As there is a need for model estimation, Maximum A Posterior (MAP) solutions are the most common choice when MRFs are used which makes it possible to create a probabilistic framework for solving the problems. As inferring MAP-MRF proved to be a difficult task, using graph theory and max-flow/mini-cut scheme which turns the MAP-MRF problem to a graph partitioning problem proved to be very useful. Thus many methods have been devised for solving the max-flow/min-cut problem.

In case of image segmentation, the mentioned methods come together in the famous Grab-cut method [19] to make an iterative energy minimization algorithm for the task of segmentation (which is actually a binary labeling problem).

In the following chapters, this framework is used in the proposed automatic human subject segmentation algorithms.

Chapter 3

Statistical Shape Model Feedback Segmentation (SSFSeg)

3.1 Introduction

As mentioned in Chapter 1, the following problems exist for human subject segmentation that is dealt with in this thesis.

1. Shape variations due to human body movements
2. Shape variations due to human wearing different clothes
3. Variation in color and texture of the clothing
4. Complexity of the scene

In this chapter, we try to solve the problems related to the shape variations and scene complexity. From many methods and algorithms available in literature, in recent years, interactive methods utilizing Graph-cut based segmentation framework have become popular due to their relatively good segmentation results. The main benefit of these kinds of methods is that they try to automatically learn the needed parameters to differentiate between the foreground object and the background. But as the name indicates, this is mostly made possible by relying on the user inputs for the initialization and the refinement procedures for achieving desired results. Still, if the need for user interaction could be removed from these methods while keeping their performance, not only our work would become much easier, the achieved unsupervised method would be applicable to a wider range of applications (both in image and video processing). Again, because of the reliance on the user, many of interactive methods do not make any kind of assumption about the object in the foreground or the background, and try to segment the object based on its feature(s) (usually color or intensity).

In Chapter 1, it was also mentioned that using a model or other prior information about the object to-be-segmented can improve the accuracy of the segmentation system considerably. Considering what has been discussed until now, it will trigger some questions in mind:

1. What would happen if we make these methods object specific (in this thesis a human subject)?
2. What kind of model/prior information can be useful in these kinds of tasks?
3. How the user interaction could be reduced to the minimum or (even better) removed from the segmentation process?

These questions become the main aspiration of the work presented in this chapter. As for the first question, as mentioned before in the summary of Chapter 1, using some kind of modeling can be very effective on the segmentation results. This will also be explained, presented, and proved later on through this chapter. For the second question, since a human body has an articulated structure, incorporating information about the changes in the shape of the body to the segmentation system can affect the final segmentation result considerably. From different methods of modeling available, we have chosen the Statistical Shape Model (SSM) by Cootes et al. [113] to encode the variations of body/cloths into an statistical framework. SSMs use an statistical modeling framework which is relatively simple, yet powerful. Their implementation is also easy and its new shape generation is fast. But their main benefit is that by encoding the variations into Eigen-space, they can efficiently reduce the number of parameters needed for shape generation and also it is possible to generate shapes which is not included in the original encoded variations (it is possible to interpolate between the data input to SSM and generate some shapes that is not included in the input set). This will allow us to make a system capable of encoding the shape variations and refining the created model at a relatively low cost (with respect to the CPU). The last question can be answered through using a human body detector for finding the location of the body and trying to automatically initialize the segmentation process in the area indicated by the detector as it will be explained later on. For segmentation, since the Graph-cut framework has good segmentation accuracy, it is preferable.

But the method used in the original Graph-cut paper by Boykov et al. [18] demands that the user must select some pixels as seeds for foreground and background. It also tries to segment the object based on grayscale histograms and intensity distance between image pixels. As a result, the Grab-cut [19] framework which is the upgraded version of that and uses color information and iterative energy minimization algorithm, will be used here. Aside from good segmentation accuracy and being well-known, the method by Rother et al. has already reduced the user interaction to selecting a polygon (Rectangle is the most famous) around the object-of-interest. Though if the segmentation result is not satisfactory, the user is forced to input some corrections and repeat the process until obtaining a satisfactory result. Aside from that, the iterative energy minimization will make it possible to adjust the amount of details expected to be obtained from the segmentation step or the amount of refinement user can provide to the input shape information which is sufficient for the work in this chapter.

As a result, the main matter that will be explained from here on in this chapter will be the following points:

1. Using a human body shape model as prior information for segmentation
2. Implementing a feedback system for improving the segmentation accuracy
3. Adding a normalized distance function for achieving more precise segmentation

As we will see in the experimental results, by using a human body shape model, the ability to cope with various body deformations is achieved, resulting in more robustness and accuracy compared to the conventional methods. We will also use the generated shapes from a pre-trained SSM as prior information for the Grab-cut segmentation stage. To refine the model for achieving best possible results, also a feedback system based on SSM shape generation is proposed. It introduces a coarse-to-fine shape generation procedure which refines the generated shapes step-by-step. This makes the segmentation results more accurate at each step, thus achieving a segmentation system which is more robust and has more accuracy than Grab-cut and has also the automatic segmentation capability.

The rest of this chapter is organized as follows: Section 3.2 will explain the details of the proposed method. Some experiments have been performed for testing the validity of the proposed method which will be presented in Section 3.3. There would be a discussion about system implementation and parameters selection in Section 3.4, and finally method presented in this chapter will be summarized in Section 3.5.

3.2 Statistical Shape Feedback Segmentation

3.2.1 Main Idea

Two ideas proposed in this chapter are as follows:

1. Using the knowledge of human body shape as prior information for human segmentation
2. Implementing a feedback system with a coarse-to-fine shape generation schema which helps the system achieve more accurate results

As for the Idea 1, it is understood that in object segmentation, introducing a general segmentation which can segment any given object-of-interest would be very difficult, so selecting a subject for segmentation would make it much easier.

By knowing the object to be segmented, we can use various types of information as priors for modeling and segmentation. In case of human being, due to the deformability

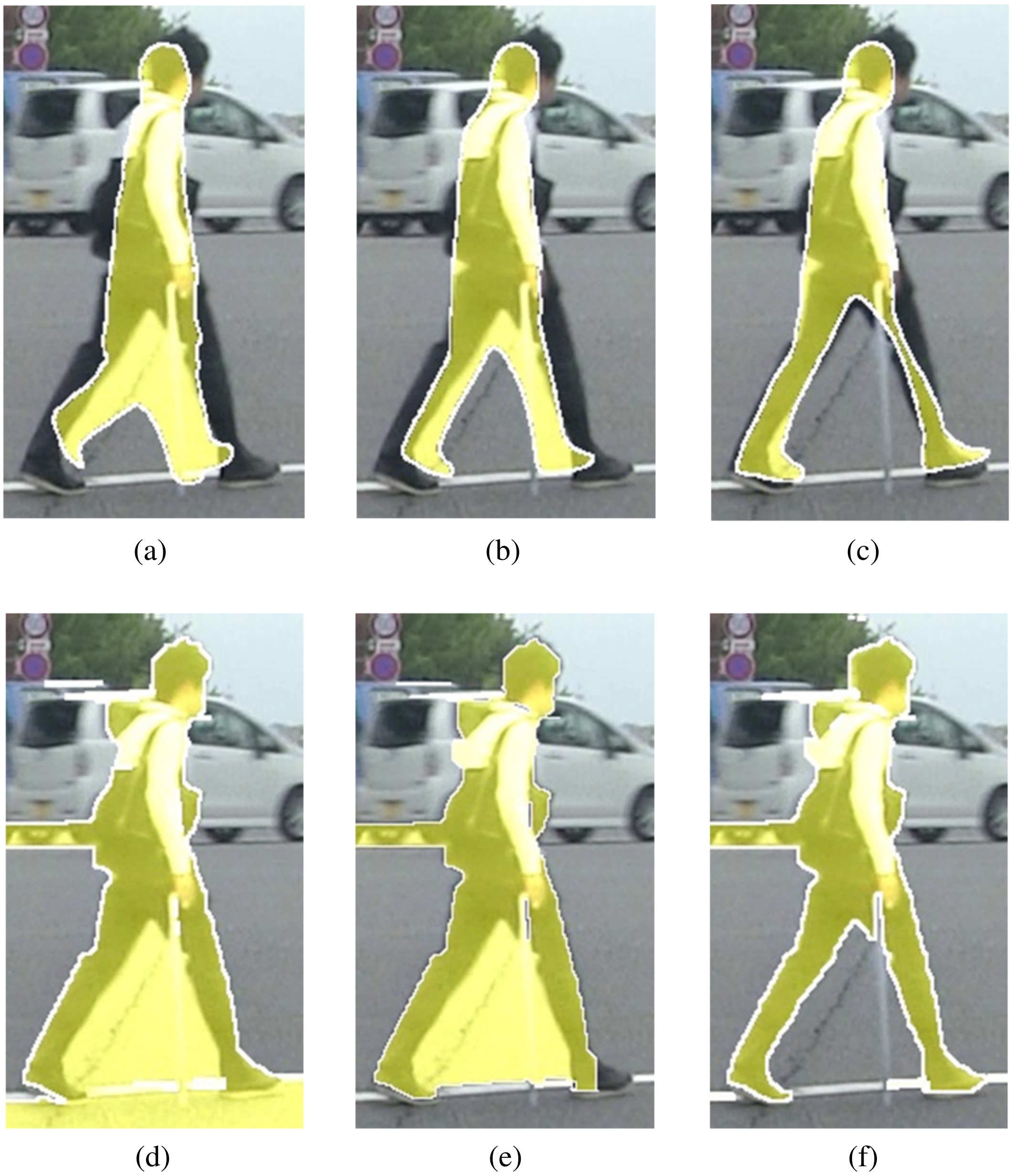


FIGURE 3.1: Example of step-by-step mask refinement. (a) Best mask selected after local refinement of the shapes generated in the initial SSM shape generation. (b) Best mask which is selected after local refinement of shapes generated using parameters of (a) and average shape model. (c) Best and final local mask selected from shapes generated at third model refinement step using (a) and (b) parameters. (d) ~ (f) Results of segmentations using (a) ~ (c) respectively.

of the body itself, and also various color and shape changes due to different types of clothing, this task is very difficult but not impossible. At least modeling the human body shape is possible because even if the body is highly deformable, since there are some physical constraints on it, the degree of the shape variations is limited. So, it is possible to model the shape changes in a mathematical way. If we can find a model which is relatively simple and can model the shape changes to an acceptable degree, we can use this model as a prior for segmentation. This would give us the possibility of segmenting the subjects with more accuracy.

This idea is exploited in the segmentation system introduced in this chapter. As one of the ways to model the shape, the SSM method [113] was chosen to model the human body shape and use the generated samples from a system, trained with real pedestrian silhouettes, as a basis for human subject segmentation. The aim is to use the flexibility of the SSM algorithm for generating new shapes in addition to the segmentation accuracy of the Grab-cut and propose a system which can segment human subjects automatically and accurately.

Although it is possible to generate various shapes with SSM and use them to improve the segmentation result, still there is no guarantee that the generated shape matches the actual subject to be segmented, and as a result, just by generating shapes, we might not obtain the desired result. So, there is a need for a way to tell the shape generation process that the shapes which are being generated are having good effect in the segmentation or not. This is where Idea 2 shows its usefulness. A feedback system can help a lot by providing a way of knowing if generated shapes are good or not. It also can help speeding up the shape generation by reducing the number of shapes that is generated each time, i.e. instead of generating a lot of shapes at once, first we can generate some rough shapes and by using feedback, refine it until we obtain the desired result. The effect of using feedback and also shape refinement is presented in Figure 3.1.

The proposed method is called “Statistical Shape Feedback Segmentation” and in the rest of the thesis, it will be referred to in abbreviated form “SSFSeg”. The general process flow of the system is shown in Figure 3.2.

Using the mentioned ideas and by modifying the Grab-cut segmentation method, the proposed system which can be summarized in the following procedures:

- **SSM generation step**
  - Some new samples based on the training data are generated.

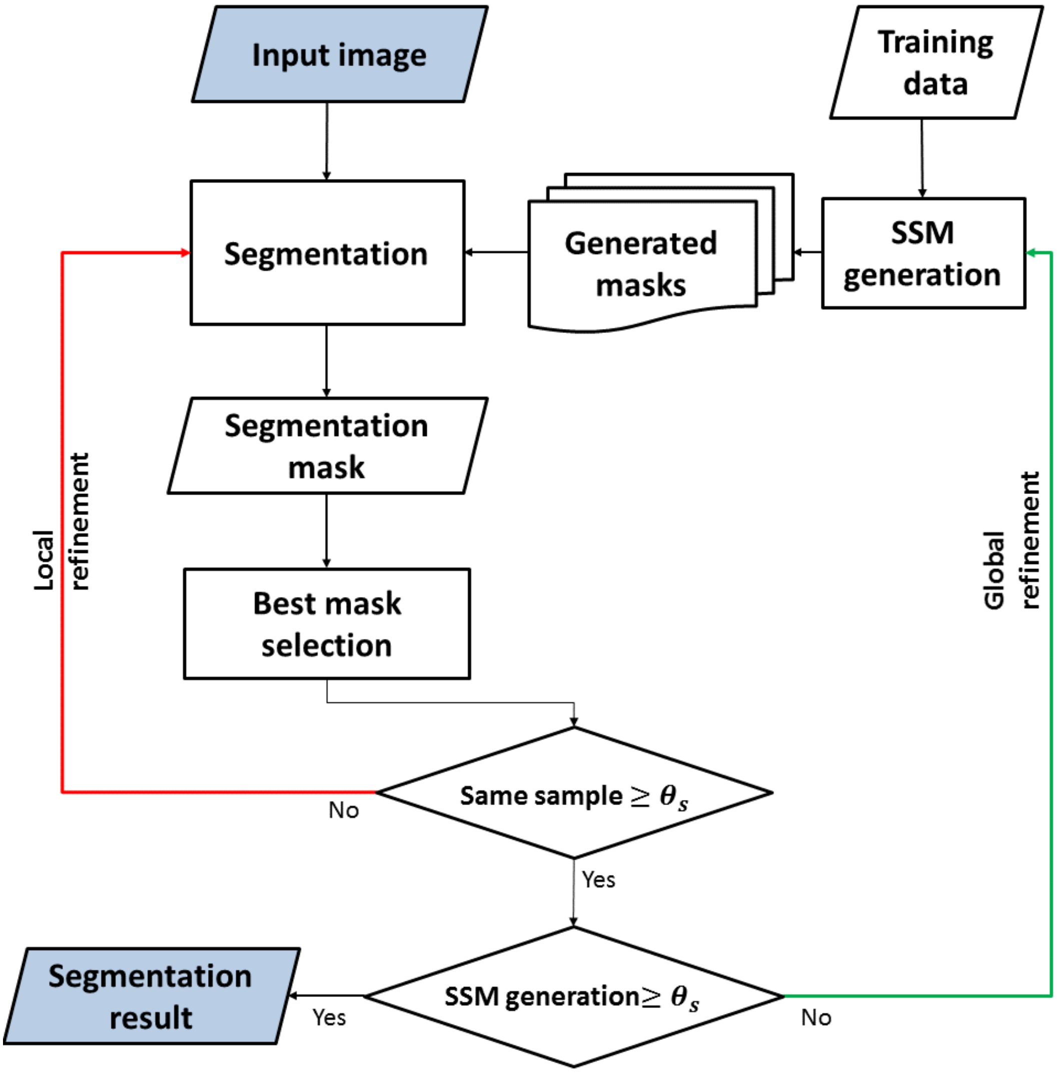


Figure 3.2: Process flow of the proposed SSFSeg method.

- **Mask generation step**
  - The selected sample is converted into a tri-map (refer to Section 2.3.1).
- **Segmentation step**
  1. An image containing a human subject is input.
  2. Labels are assigned to each pixel based on the generated mask from the SSM generation step.
  3. For each pixel in the unknown region, a GMM for foreground and a GMM for background are assigned.

4. From input data, GMM parameters are learned.
5. Segmentation is done using the max-flow/min-cut algorithm.
6. Repeat from Step 3 until convergence.

- **Local refinement process**
  - Repeat the segmentation step until a good local sample is found.
- **Global refinement process**
  - If the segmentation result stabilizes, terminate the procedure and show the result, otherwise, start over from the SSM generation step.

In the rest of this section, each of the items above (SSM generation, Mask generation, Segmentation, Local refinement, and Global refinement) will be explained in more details.

3.2.2 Statistical Shape Model (SSM) Generation

Although there are many ways to model a human body, like Active Shape Models (ASM) or Active Appearance Models (AAM), and so on, here Statistical Shape Model (SSM) is used as the method for encoding the training samples into a mathematical model for the proposed segmentation system. SSMs lay a very useful ground for encoding the changes in the shape of the object through Eigen-space analysis. By analyzing the changes in the Eigen-space and finding the dominant parameters responsible for changes in the shape and changing them, it is also possible to generate some new shapes not included in the actual training dataset. This is a very useful capability which has been taken advantage of in this chapter.

The first step starts with generating some new shapes based on the training dataset using the conventional SSM method, first introduced by Coots et al. [113]. This method gives us the capability of defining the shape of objects in a mathematical manner and use this representation for further works.

For making the model, first, some training images are segmented manually, creating a binary silhouette image based on the desired foreground object (the foreground object can be anything, in our case, it is human but other types of objects with varying shapes can also be targeted). After that, the boundary of each object in the training set will be turned into a vector by selecting some points around the boundary. The shapes can be aligned beforehand or be aligned as described by Cootes et al. [113]. Thus, for each image, there will be a vector with $2n$ points like

$$\mathbf{x}_i = [x_1, y_1, ..., x_n, y_n]^T \tag{3.1}$$

The mean model for the shape domain can be calculated as:

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i \tag{3.2}$$

Based on these, the covariance matrix can be calculated as:

$$\boldsymbol{S} = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{3.3}$$

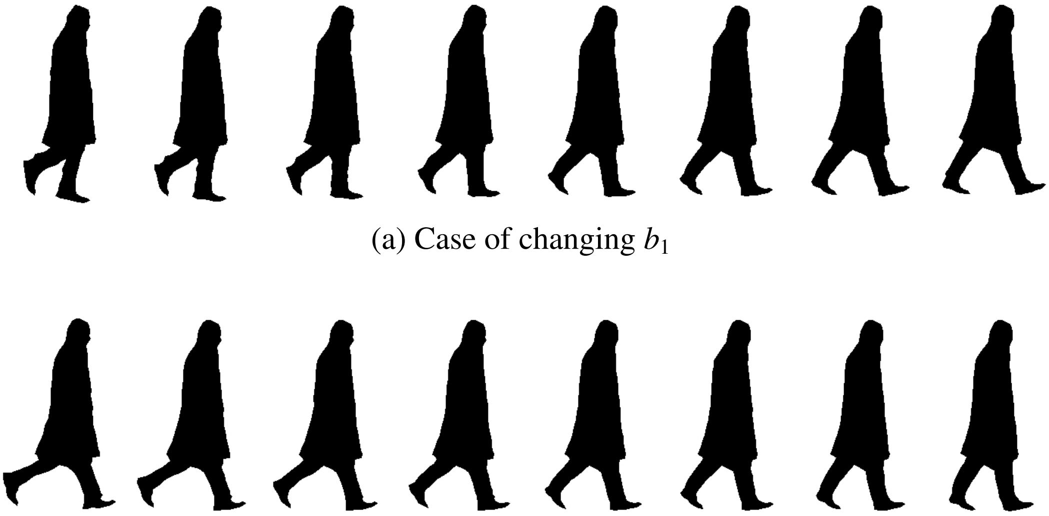
By analyzing this $2n \times 2n$ matrix, it is possible to calculate its eigenvalues (λ_i), corresponding eigenvectors ($\mathbf{p}_i$), and selecting a small set of them, we can generate new samples approximating the original training samples with the following equation.

$$\mathbf{x}_{\text{new}} = \bar{\mathbf{x}} + \boldsymbol{P}\mathbf{b} \tag{3.4}$$

Here, matrix $\boldsymbol{P}$ is made by setting the selected eigenvectors as columns, and $\mathbf{b}$ is a vector of weights like

$$\boldsymbol{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_t] \tag{3.5}$$

$$\mathbf{b} = [b_1, \ldots, b_t] \tag{3.6}$$



(a) Case of changing b_1

(b) Case of changing b_2

FIGURE 3.3: Some samples generated with SSM.

As described in the work by Cootes et al [113], a suitable limit for the weights can be defined as:

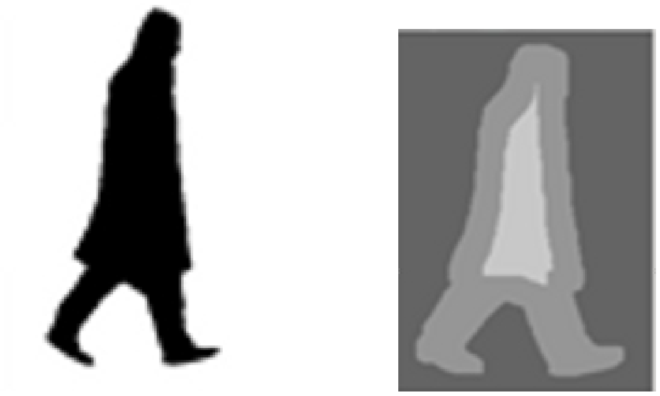
$$-2.5\sqrt{\lambda_k} \leq b_k \leq 2.5\sqrt{\lambda_k},\ k \in [1, \ldots, t] \tag{3.7}$$

In Figure 3.3, some samples generated by changing the values of b_k are presented. Note that each set of samples is created by changing just one value, for example, $\mathbf{b} = [b_1, 0, \ldots, 0]^T$.

3.2.3 Mask Generation

After new shapes from the SSM generation step are obtained, they need to somehow be used as prior information for segmentation.

For this, a tri-map of labels is made for the initialization of the segmentation step and also for further segmentations in local and global refinement stages. For labeling purposes, three types of labels are defined as



(a) Generated mask (b) Resulted tri-map

FIGURE 3.4: Converting a generated sample to a tri-map of "Foreground" (light gray), "Probably foreground" (gray), and "Probably background" (dark gray).

- Foreground: Tells the system that this part is definitely foreground (object-of-interest) so it must be included in the final segmentation result. The system must try to find other object parts based on this selection.

- Probably foreground: Tells the system that the probability of this part belonging to the foreground is more than being part of the background. It is possible to change this label to other labels so all of the pixels labeled with this might not be present in the final result.

- Probably background: Similar to the "Probably foreground", but defined for background pixels (this time the probability of this part being part of background is higher).

To make the tri-map, an erosion binary operator is used to create the main part of the mask. This part is labeled as "Foreground" in the tri-map. Since a human subject would not have the same shape as the generated mask, we should somehow tell the system to search the image in an area more than the one indicated by the mask itself. This is done by dilating the generated mask and labeling that part as "Probably foreground". Aside from these two parts, the rest of the image will be labeled as "Probably background" in the final tri-map. Figure 3.4 shows an example of a generated mask and the tri-map created from it.

Table 3.1: Different γ parameters and their effect on the final segmentation result.

Gamma Parameters	**Error** (%)
$\gamma_0 = 0.06, \gamma_1 = 100$	18.07
$\gamma_0 = 0.07, \gamma_1 = 75$	**16.34**
$\gamma_0 = 0.07, \gamma_1 = 100$	18.46
$\gamma_0 = 0.08, \gamma_1 = 100$	18.07
$\gamma_0 = 50, \gamma_1 = 0$	21.57

3.2.4 Segmentation

Now that shape models are generated and one of them has been converted into tri-map, the segmentation procedure can be started. Here, the original Grab-cut by Rother et al. [19] is modified by adding a distance penalty to the γ parameter which controls the smoothness term of the energy model used in Grab-cut forcing it to value the points near the boundary of the input mask more than the other parts. This usually leads to more accurate segmentations based on the experiments done.

As a result the constant γ value in the smoothness term of Equation 2.35 becomes a variable parameter which assign each pixel of the image with a value relative to the Euclidean distance of that point to the nearest point on the boundary of the mask input to the segmentation stage. Thus the smoothness term in Equation 2.35 in Chapter 2 can be rewritten as:

$$V(\underline{\alpha}, \mathbf{z}) = \gamma(m) \sum_{(m,n)\in\mathbf{C}} \frac{\exp^{-\beta\|z_m - z_n\|^2}}{\mathrm{dist}(m,n)} \tag{3.8}$$

in which

$$\gamma(m) = \gamma_0 + \gamma_1 \times \mathrm{dist}(m, m') \tag{3.9}$$

again the equation holds when $\alpha_n \neq \alpha_m$. m' is the nearest point on the boundary of generated mask to point m in the image and $\mathrm{dist}(m, n)$ is the Euclidean distance function. The resulted map is also normalized so that the difference between points does not become dominant in the smoothness term. The values for γ_0 and γ_1 are also selected based on the experiments with different sets of images. Table 3.1 shows how changing the values of parameters affects the final segmentation accuracy.

3.2.5 Local Refinement

After segmentation, the resulted output for foreground will be compared with the input prior mask and the error rate will be calculated. Also, the output will be compared with other generated samples and the most similar one (the sample whose error rate is less than the others) would be selected for the segmentation process. This step would be repeated until the system converges to one of the samples (the same sample is selected repeatedly; more than θ_s times).

3.2.6 Global Refinement

After the local refinement process, parameters for the current and the previous samples with the least error rate are calculated and based on that, a new set of samples (again, N samples) are generated. The same process is repeated for finding samples with the least error. The whole process of sample generation and image segmentation will be repeated for more than θ_g times, and the final result would be presented to the user. Figure 3.1 illustrates an example of how the feedback system refines the selected masks at each refinement step.

Since our assumption is that the system cannot use any kind of user provided data, the generated sample is used as the ground truth for segmentation provided that the desired object should be similar to the provided mask to some degree.

3.3 Experiments

In this section, results of different experiments for validating the SSFSeg method are presented.

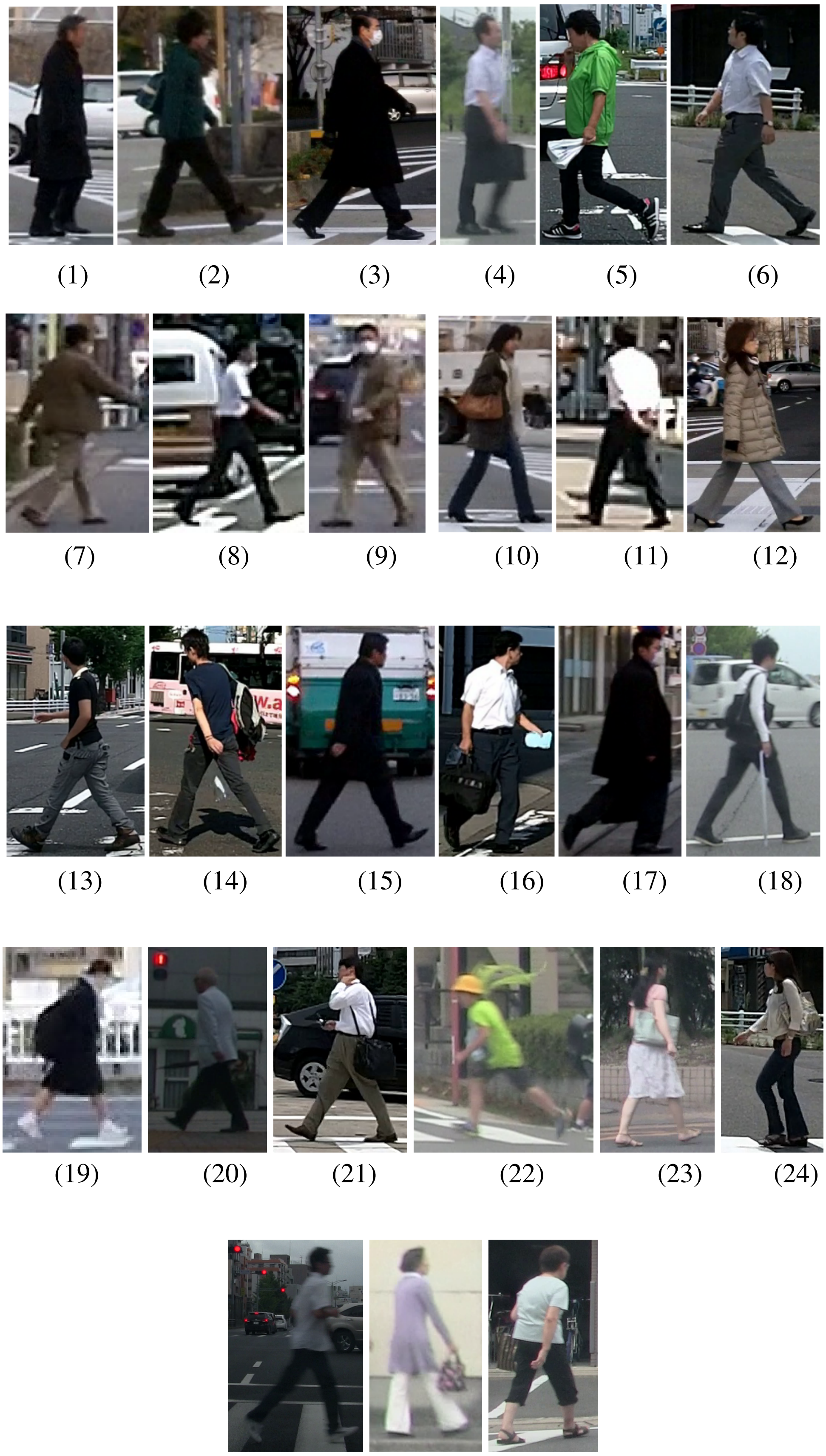
3.3.1 Dataset

Three datasets have been used for our experiments here.

The first dataset used for testing the system here, is a private set of 180 images from different human subjects (full body) in various situations. The images in the dataset are extracted from high definition recorded video ($1,280 \times 720$ pixels) taken by a camera installed behind the windshield of a vehicle. It contains footage of pedestrians crossing the street or walking in pathways. Videos are taken in bright, normal, and dark places. All images are RGB color containing pedestrians with different sizes from 47×80 pixels to 378×618 pixels. The images are all taken in the day light thus, night time images are not included in neither training nor testing experiments. Some samples of the test dataset are presented in Figure 3.5, with their size and processing time in Table 3.2.

The second data-set has been created from Caltech Pedestrian Detection Data-set [3, 114] which is a famous data-set as a benchmark for pedestrian detection methods. The video recording setup is the same as in our dataset but, Instead, the video size is in VGA (640×480 pixels) which implies that the pedestrian sizes are smaller. Since the quality of the images and pedestrian sizes are not very good in this data-set, 100 human subjects with a height more than 50 pixels were selected randomly. As in the first private dataset, these images are recorded in the day light with no night time images included.

The third dataset is also a private dataset used for training the SSM shape model generator. For this dataset, 60 samples are used for training the SSM model which are not included in either of the test datasets. All samples are turned into binary hand-segmented silhouettes of real pedestrians. These pedestrians are selected from images taken by in-vehicle camera in the same way as test datasets. The training image size is set to 371×540 pixels and all training silhouettes are scaled to the same size, keeping their original aspect ratio. Please note that this set is common for all tests.



(1) (2) (3) (4) (5) (6)

(7) (8) (9) (10) (11) (12)

(13) (14) (15) (16) (17) (18)

(19) (20) (21) (22) (23) (24)

(25) (26) (27)

FIGURE 3.5: Some samples from the testing dataset containing 180 pedestrian images.

TABLE 3.2: Size and processing time for the sample images in Figure 3.5

Image No.	Size (pixels)	Time (sec.)
1	94×216	16.83
2	150×217	33.32
3	150×241	41.29
4	93×234	13.30
5	207×393	151.55
6	198×320	82.06
7	83×127	6.36
8	72×104	5.55
9	56×105	10.15
10	116×221	16.90
11	59×100	4.26
12	150×242	31.70
13	246×412	169.59
14	204×335	133.09
15	142×221	23.93
16	147×293	48.86
17	150×282	35.66
18	162×279	63.39
19	70×104	7.44
20	121×210	24.47
21	179×276	49.80
22	178×204	41.05
23	167×300	76.15
24	154×311	7.75
25	368×580	760.06
26	98×185	14.92
27	205×399	150.01

3.3.2 SSM Sample Generation

Since real human body is used as the basis for training, chance of generating more realistic priors for the segmentation stage increases, thus the final segmentation result would become more accurate since we can include more realistic shape variations.

At each stage of sample generation, $N = 50$ samples are generated. For the first segmentation, the mean shape is selected as the start point. For the criteria to terminate the segmentation and the generation process, experiments show that if the parameters are set to $\theta_s = 5$ and $\theta_g = 3$, as it can be viewed from Table 3.1, usually desired results would be obtained.

3.3.3 γ Parameter Selection

It can be seen in Table 3.1 that if parameters are set to $\gamma_0 = 0.07$ and $\gamma_1 = 75$, the best results are obtained. So, in all of the tests for the proposed method, these values have

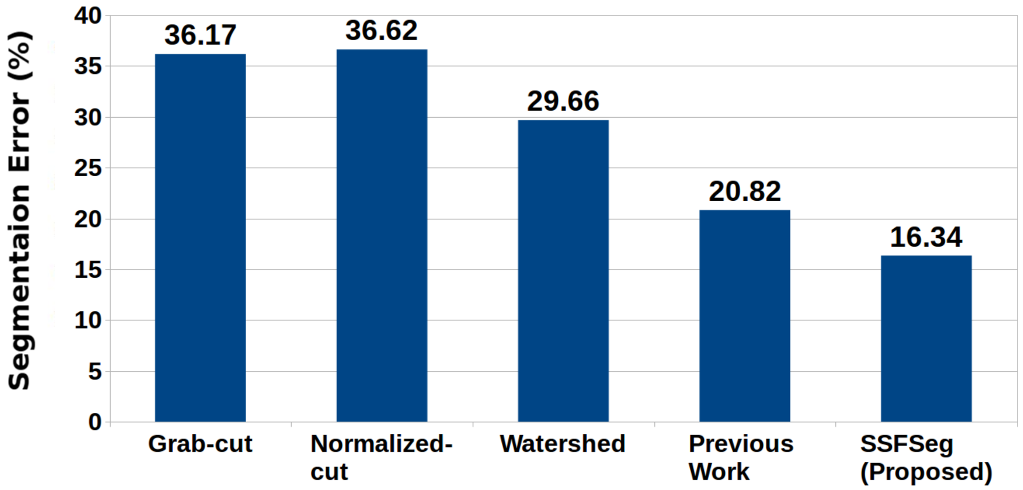


Figure 3.6: Comparison between the proposed method and other segmentation methods using the private dataset.

been selected for $\gamma(m)$. Thus Equation 3.9 becomes:

$$\gamma(m) = 0.07 + 75\,\mathrm{dist}(m, m') \tag{3.10}$$

3.3.4 Results

Comparison is performed between the original Grab-cut segmentation [19], Normalized Cut (N-cut) segmentation [65], Watershed [115] segmentation and the proposed SSFSeg method. The segmentation error of the methods is calculated based on the number of pixels that have been miss-segmented as foreground or background in comparison to the ground-truth provided by manual segmentation of the desired object. Thus

$$\text{Error (\%)} = \frac{\text{FN} + \text{FP}}{\text{Number of pixels in the image}} \times 100 \tag{3.11}$$

where FN is the number of foreground pixels segmented as background and FP is the number of background pixels segmented as foreground.

For Grab-cut and Watershed segmentations, the code provided by OpenCV [116] open source library, and for Normalized-cut segmentation, the code provided by Cour et al. [117] were used.

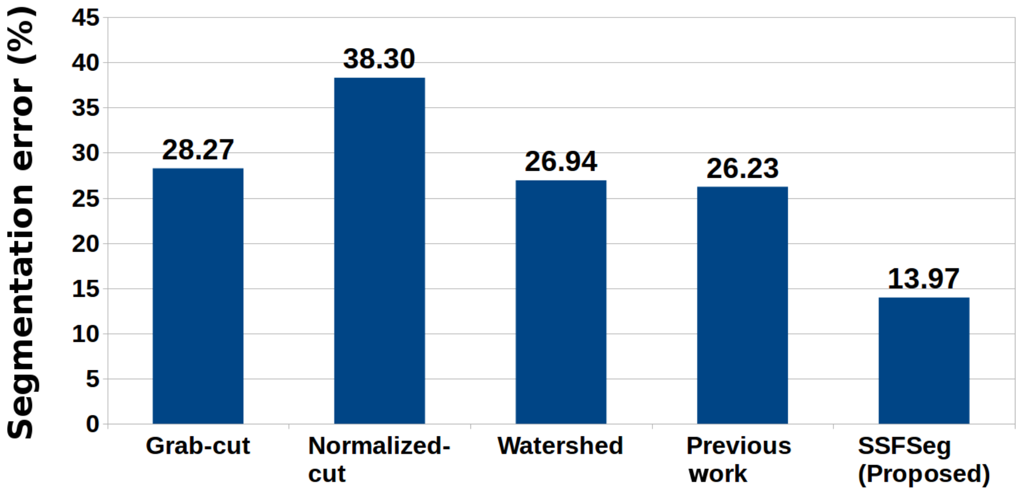


FIGURE 3.7: Comparison between the proposed method and other segmentation methods using images from the Caltech pedestrian dataset [114].

Figures 3.6 and 3.7 show segmentation error from using the proposed method and comparative methods. As it can be seen in the images, segmentation error is significantly decreased compared to other methods (cut by half comparing to the Grab-cut and Normalized-cut).

There is also Figure 3.8 which illustrates how many of images are segmented with more than a specific accuracy. For example, from the figure, it can be seen that in the proposed method, 168 images out of 180 images in the dataset are segmented with accuracy more than 70% while this number is 40 for Grab-cut, 53 for Normalized-cut, and 89 for Watershed.

Figure 3.9 shows the segmentation results by the proposed system and its comparison to other methods. As it can be seen, the results have improved noticeably compared to other segmentation methods.

3.4 Discussion

Some questions might arise about how parameters are selected for the system and how changing the values selected for the system affects the segmentation result.

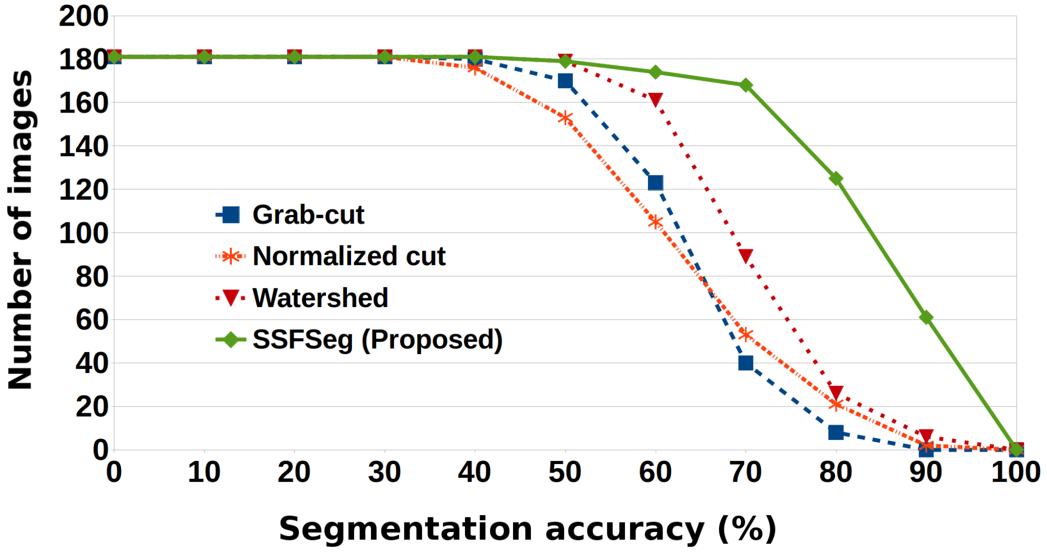


FIGURE 3.8: Performance of the proposed and the comparative methods. Number of dataset images that were segmented with accuracy over 70% by the proposed SSFSeg method is much higher compared to comparative methods.

As for θ_g and θ_s, Figure 3.10 shows the results of repeating the SSM generation step from 1 to 10 times and local refinement from 1 to 10 times. As it can be seen, the best result is achieved when parameters are set to $\theta_s = 5$ and $\theta_g = 3$.

Although in the overall process, changing these values affect the final segmentation result within a 3% range, finding the optimal parameters helps us avoid wasting time for unnecessary shape generation and local refinement. Some experiments are also performed for observing the effect of changing the γ factor in the smoothing term of the Grab-cut segmentation stage. The results of these experiments can be seen in Table 3.1. As the table shows, in the experiments when the γ parameters are set as in Eq. 3.10, the segmentation error decreases to its minimum value (16.34% in the segmentation experiments).

It should be noted that using feedback in the system can affect the final result significantly. It makes it possible to generate new shapes based on the segmentation result which has two benefits. First, the number of shapes that has to be generated at each step is decreased to a small set of 50 images, and second, it is possible to refine the generated mask to become as similar as possible to the segmentation result, thus improving the final result. The effect of using feedback in the system can be seen in Figure 3.11.

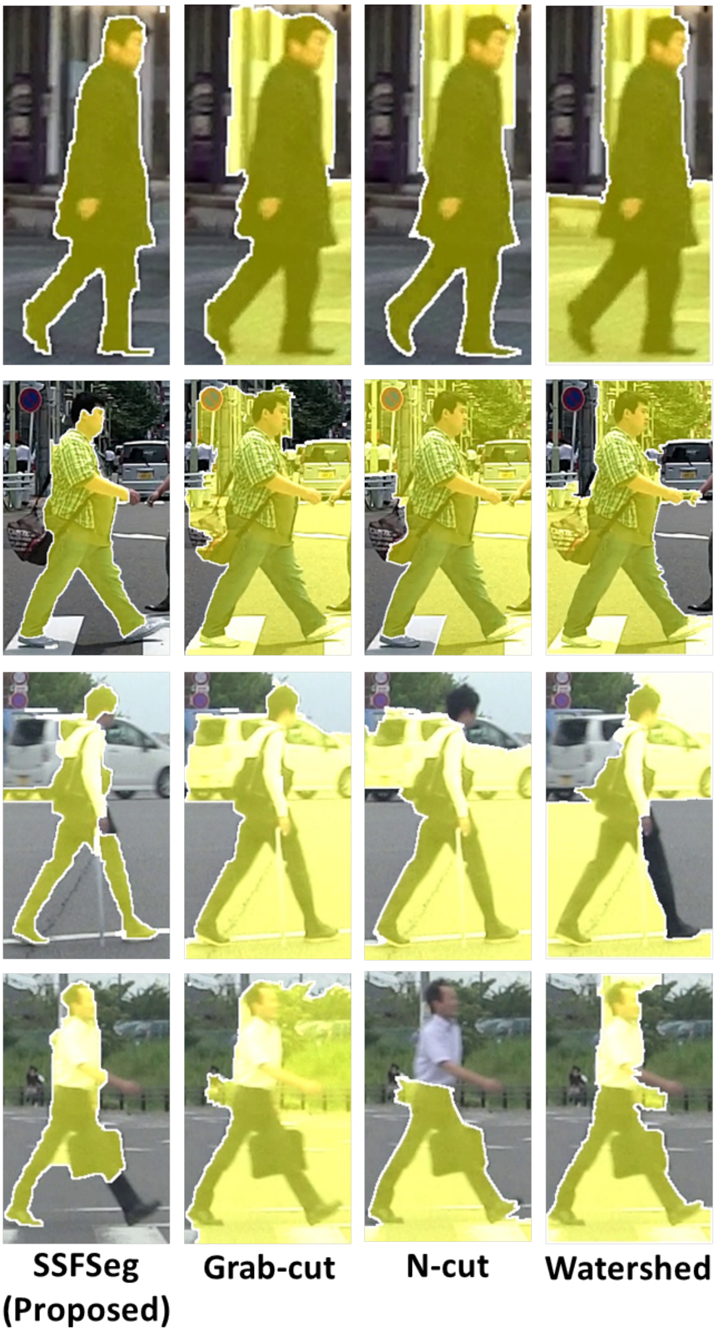


Figure 3.9: Example of image segmentation results. Results are overlaid on the input image with a yellow colored mask.

Table 3.2 shows the time required for processing the images in Figure 3.5. The system has been implemented in C++ and is not optimized at current stage. Also it uses single thread for computation purposes. Note that by code optimization, it should be possible to reduce the time consumption significantly.

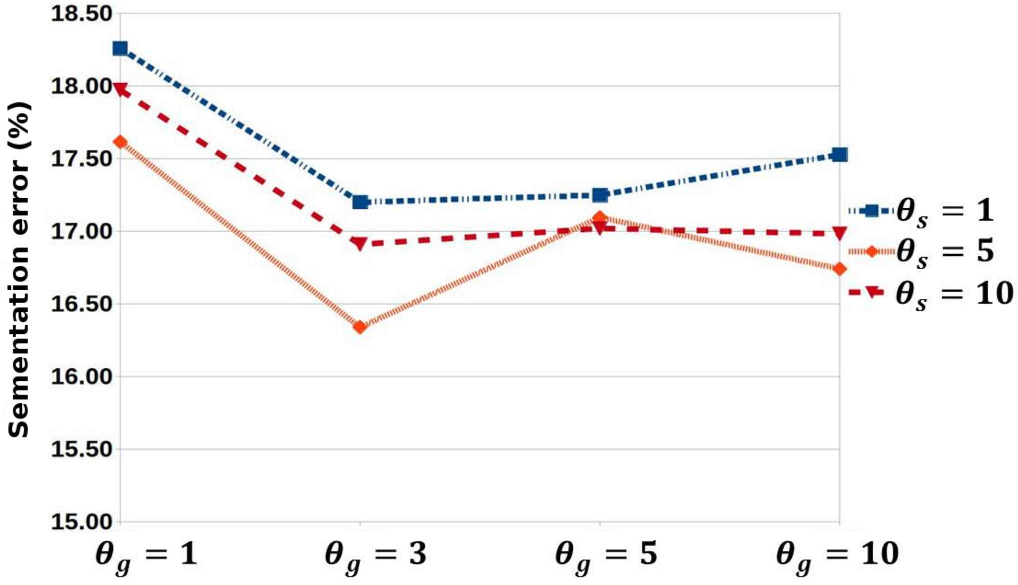


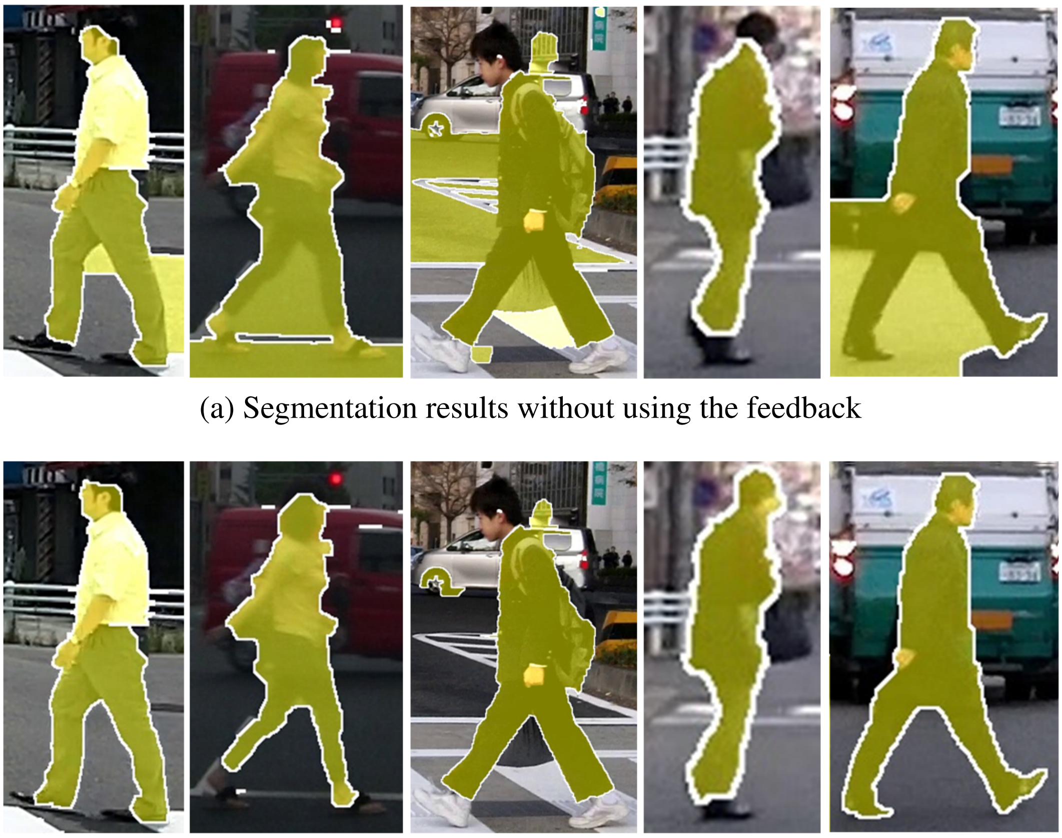
FIGURE 3.10: Relation between θ_s and θ_g, and the effect of changing their values on segmentation error.

Still there are also some miss-segmented cases as shown in Figure 3.12. The problem in the first row is mainly because of the similarity between foreground and background colors. Meanwhile, the second row shows the miss-segmentation because of wrong seed selection.

It is also good to note that the proposed system uses and generates full body silhouettes at the SSM stage so it does not consider the case of occlusions. The method explained in this chapter expects the output of a human detector algorithm (e.g. [118] and [119]) as an input. Therefore if there exists more than one human subjects in the image, all detected human subjects can be segmented by applying the proposed method for each of them separately.

3.5 Summary

In this chapter, an automatic pedestrian segmentation method has been presented which can perform the task accurately. The main idea is to make the process automatic by using the SSM model generation algorithm to generate some prior masks for the Grab-cut



(a) Segmentation results without using the feedback

(b) Segmentation results using the feedback

FIGURE 3.11: Result of omitting and using feedback on the segmentation accuracy of the proposed method.

segmentation step instead of asking the user to identify the background and foreground seeds.

From the problems mentioned in the Chapter 1, some were selected and presented in the introduction of this chapter to be solved. Based on that, a method was proposed to solve the problems of shape variation and complex background together. For this, the possible variations of the shape are encoded in an SSM model and then a feedback system will monitor the segmentation results and refines the generation model to get the best possible results. Combing the SSM shape generation and feedback not only makes the segmentation result more accurate, but it also removes the need for user interaction altogether. As a result, when used in combination with a human body detector a fully automatic human subject segmentation system is achieved. Also, since the proposed method uses a single-frame monocular image information, it is possible to easily adapt

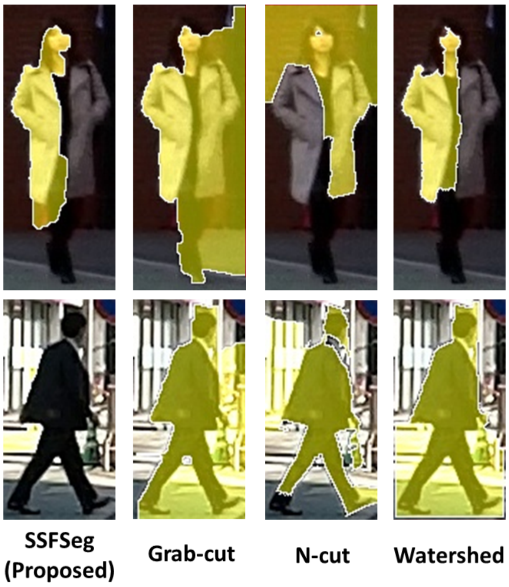


FIGURE 3.12: An example of miss-segmentation in the proposed method and equivalent segmentation in comparative methods.

it to multi-frame/camera human segmentation scenarios too (Using multi-frame/camera information might also improve the accuracy and reduce the computation time, and also widen the range of applications the proposed methods can be applied to).

It should be mentioned that even if the SSFSeg method can perform the segmentation automatically and sometimes with better accuracy in comparison with the Grab-cut method, still there are some problems that have to be solved so that it becomes applicable in real situations.

1. Since the Grab-cut just uses color features for foreground and background segmentation, if the color distribution between an object and its background is not very different, we will not be able to obtain a satisfactory result.

2. The method proposed here works on the input frames containing the full human body which is provided by a human detector like HOG human finder or any other detector. This means that if the detector finds multiple human subjects in an image, they will be segmented using the proposed method. But it also means that depending on the detection accuracy, there might be cases in which a human subject is not detected and as a result will not be segmented either.

3. To reduce the complexity of the shape model used in the proposed method, the case of occlusions are not considered here. This means that the shape model used in this chapter can only generate full human body shapes and cannot cope with the human subjects that are occluded. As a result, the system might be able to produce satisfactory results in case of minor occlusions, but the segmentation accuracy will be affected considerably in cases with major occlusions. This is also one of the problems that has to be solved.

4. Another problem is the computation time needed for converging to the best segmentation result. As mentioned before, the proposed method uses a coarse-to-fine scheme for achieving the best model to segment the human subject. This means there is a need for segmenting the image multiple times to obtain the best result which is time consuming. This problem might be solved by optimizing and multi-threading the code and running that on GPU instead of the single-thread code on CPU.

Chapter 4

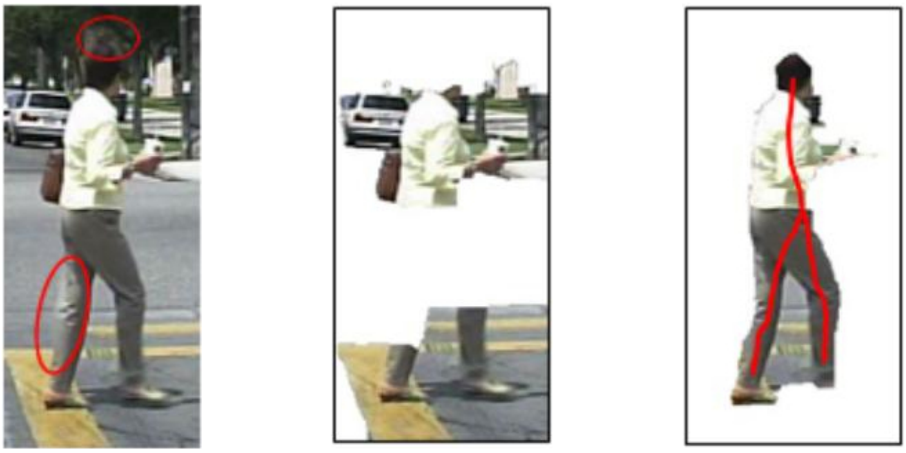
Human Segmentation Using Super-pixels & Probability Map

4.1 Introduction

In Chapter 1, the following problems were mentioned to be the focus of this thesis:

1. Shape variations due to human body movements
2. Shape variations due to human wearing different clothes
3. Variation in color and texture of the clothing
4. Complexity of the scene.

From the problems mentioned above we have tried to solve the problems related to the shape variation and the complexity of the scene in Chapter 3. As mentioned there, recent interactive methods have good segmentation accuracy but they need user interaction. Another problem is that many of the Graph-cut based methods use color (or intensity) as their main feature for discrimination between the foreground and the background pixels which has proved to be not a very fault tolerant feature to work with. Also, as mentioned in the summary of the previous chapter, the coarse-to-fine model



(a) Input image

(b) Segmentation without seeds

(c) Segmentation with seeds

FIGURE 4.1: Grab-cut failing to segment the human subject due to similarity of the color between body parts and background color.

refinement is time consuming which might not be a problem for tasks that time is not of concern, but in applications like driver assistance systems, real-time performance is one of the crucial factors. Another problem with the previously proposed method is that although the shape model refinement can produce good initializations, generating full details of the body becomes a very difficult task. Also, there is no guarantee that the produced shape would match to the actual input human subject completely. This, sometimes, might also lead to miss-segmentations that should be avoided, if possible.

As a result, in this chapter, a solution to the remaining unsolved problem which is the variations in color/texture of the human subjects is proposed. In addition, the proposed method in Chapter 3 is improved by reducing the computation time through removing the coarse-to-fine process and need for initialization steps from the proposed method. In this chapter, the problem of need for user interaction is solved through using a human body probability map for selecting human body regions from super-pixels generated from an input image. The problem of color feature failure is also tackled by applying a texture feature in the process of super-pixel generation. The main problem of just using color feature is that due to the changes in the texture of the object or intensity of the environment, the system usually fails to segment all parts of the object as one. For example, in case of human body, using just color feature usually fails to provide enough information to segment different parts of the body/clothes due to changes in color or

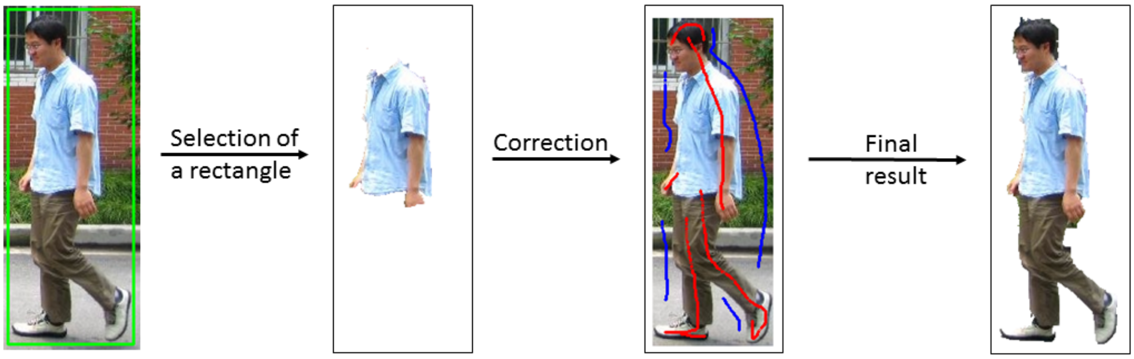


FIGURE 4.2: Example of the segmentation process by the Grab-cut method. As it can be viewed, since Grab-cut methods just use color feature for separating foreground and back ground, without further corrections by user, it fails to segment the whole human body from the image.

texture as in Figure 4.1 or Figure 4.2 (b)).

The following points are the main reasons that the combinations of human probability map, texture feature and super-pixels was employed for the task of segmentation in this chapter.

1. Using the probability map relieves us from the need for matching the shape model and the actual body, thus helps to improve the segmentation accuracy.
2. Incorporating texture information can be done in super-pixels generation. Since this kind of information which is not used by the mentioned methods, can improve the accuracy of segmentation regions that color feature alone is not sufficient.
3. Using the coarse-to-fine or iterative schemes like the one proposed in Chapter 3, will be taxing on time and computation power, so here, an algorithm to perform the segmentation in one iteration is proposed. This results in much less strain on the CPU, thus the process becomes much faster (almost by the scale of 10) while maintaining nearly the same accuracy as the mentioned methods.

The main idea comes from jigsaw puzzle game. Usually humans wear different clothes with various colors and textures. If the image is divided into regions based on their color/texture, each part of the human body then, becomes like a piece of a puzzle. So, an image can be considered as a puzzle with multiple pieces in which the human body occupies some of them. If right pieces are selected, a somewhat rough shape of the body can be obtained and by using Grab-cut, the human subject is segmented accurately. Following this idea, not only the system becomes automatic but also the accuracy of the system improves. Also, just by using the information of a single image, it is possible to achieve segmentation results with accuracy comparable to the state-of-the-art and much better than traditional methods while having a relatively simpler model. It is also good to note that this method can be used in both automatic and interactive segmentation manners.

The rest of this chapter is arranged in the following manner. Section 4.2 introduces the proposed method. Section 4.3 will provide the experimental setup and results of the proposed method compared with other methods. Sections 4.4 and 4.5 will respectively belong to discussion and conclusion of the proposed method.

4.2 Proposed Method

4.2.1 Main Idea

Before introducing the proposed method, it would be better to first provide a brief explanation about the idea behind the proposed system.

Usually, images contain different kinds of objects either as scenery (background) or the ones that are of interest (foreground). Simple objects are usually specified by a combination of its color and texture. Complex objects like humans are hard to be defined in this concept because they have different color/texture in each part. As a result, if each region in the image is mapped by its color or texture, the result would become like a puzzle with multiple pieces. So it is possible to think of this as a puzzle-game problem, in which the input image would become a puzzle with multiple pieces. The task will then become the search for correct pieces to select and find the human subject inside different pieces available at hand. As for the way to make an image into a puzzle, using methods that turn the image into some super-pixels would be the best choice. Since the aim is to segment the human body, if the puzzle is created so that the pieces

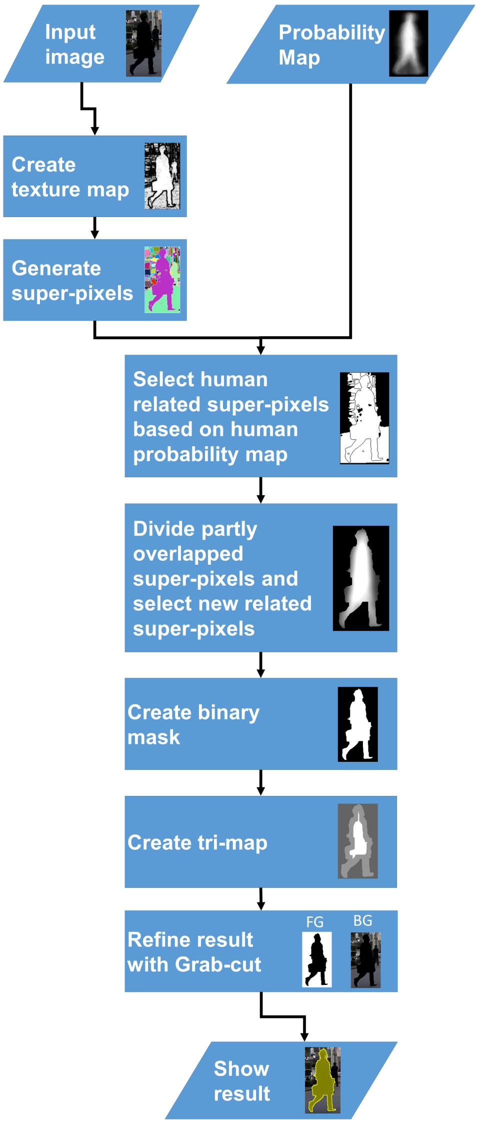

Input image
Probability Map
Create texture map
Generate super-pixels
Select human related super-pixels based on human probability map
Divide partly overlapped super-pixels and select new related super-pixels
Create binary mask
Create tri-map
Refine result with Grab-cut
FG
BG
Show result

FIGURE 4.3: Flowchart of the proposed segmentation method.

related to human body are easier to pick, it would make the task much easier. There are different methods for turning the image into super-pixels like Watershed [70], SLIC Super-pixel [92], Liu et al.'s method [120], or others. Here, Watershed algorithm [70] which is well-known and fast, is used in the super pixels generation process.

As it is mentioned before in 1.4, human subjects usually appear in different colors/textures (because of their clothing and change in the color/texture due to shadows, folding of different parts of the clothing, and so on). As a result, if the puzzle is first made from the input image in which each piece contains regions with the same texture and then try to find a human subject inside the pieces, it will be more convenient. To shape up the super-pixels to the best shape as possible, the Watershed algorithm is provided with best region candidates based on the texture of the regions. For this, a texture feature map based on the work by Zhou et al. [121] is used.

Now that the puzzle pieces have become available, the right ones should be selected as the human body. So, a human probability map is used at this stage which shows where the probable parts that the human body might exist in the image frame. For this, 64 images were segmented manually and turned into a binary image. After that, binary images were scaled to match in either width or height, while keeping their original aspect ratio. At last, resized silhouettes were added together and normalized to create a body probability map which will be input to the system. As mentioned before, this stage is an off-line procedure which means the creation of this probability map is done before the segmentation procedure is even started. In some cases, one piece of the puzzle might contain a part of the object alongside the background. In this case, it will be possible to break them into smaller pieces and select the correct parts using this probability map. Selecting the right pieces will yield the rough shape of the human subject. This rough shape can be improved to a more accurate result by feeding it as a prior to the Grab-cut framework later in the process. It is also good to note that by using this idea, it is possible to perform the segmentation in both automatic and interactive manners.

4.2.2 Overview

The basic flow of the proposed method can be seen in Figure 4.3. The inputs to the system are the image to be segmented and a human body probability map. This map represents the probability of human body parts in each part of the image. As mentioned before, the probability map used in this chapter is made by hand segmenting some human subjects, adjusting the images sizes and adding them together. It is also possible to use a trained SSM shape generator like the one in Chapter 3 to generate many shapes and then add them together to obtain the map.

The process after data input, can be briefly explained in the following steps. More details will be presented in the rest of this chapter.

- **Super-pixel generation step**

  After the images have been input to the system, the first step is to convert them to some super-pixels for further processing. Here, the Watershed algorithm [70] is used for this purpose.

- **Super-pixel selection using human probability map step**

  After super-pixels are generated, the ones related to the object-of-interest (here, the human body) should be searched for using the probability map generated before starting the segmentation.

- **Selection refinement step**

  Since there might be some parts of the subject in the super-pixels that were not selected, the one that contains a part of the object are broken into smaller ones and are checked again.

- **Result refinement step**

  The final selected super-pixels generate the main segmentation and is refined using Grab-cut.

4.2.3 Super-pixel Generation

After an image is input to the system, the first step is to turn it into some super-pixels for further processing. Here Watershed algorithm [70] is used, since it has simple yet effective formulation for image segmentation, which makes it ideal for the preprocessing stage. In addition, it can be used multiple times if needed. When provided with some seeds, it segments the image to some coherent regions based on the input. As some seeds are needed to define sources for segmentation, they are provided to the system by calculating texture features which makes the segmentation a combination of color and texture features.

For this, here, the same algorithm used by Zhou et al. [121] and Houhou et al. [51] which is based on the Beltrami representation [122], is used. The color image representation will be a 5D Riemannian manifold like

$$X(x, y) \to (X_1 = x, X_2 = y, X_3 = R(x, y), X_4 = G(x, y), X_5 = B(x, y)) \tag{4.1}$$

where x and y are coordinates and $R, G, B(x, y)$ are color values at that coordinate.

As mentioned by Zhou et al. [121] and Houhou et al. [51], textures are semi-local in nature so it is possible to use this property in our favor and create a feature map. For this, the local representation is changed to a semi-local one using a window of size $n \times n$ [pixels] around the specific location.

$$P_R(x, y) = \Big\{R(x + w_x, y + w_y) : w_x, w_y \in \Big[-\frac{n-1}{2}, \frac{n-1}{2}\Big]\Big\} \tag{4.2}$$

$$P_G(x, y) = \Big\{G(x + w_x, y + w_y) : w_x, w_y \in \Big[-\frac{n-1}{2}, \frac{n-1}{2}\Big]\Big\} \tag{4.3}$$

$$P_B(x, y) = \Big\{B(x + w_x, y + w_y) : w_x, w_y \in \Big[-\frac{n-1}{2}, \frac{n-1}{2}\Big]\Big\} \tag{4.4}$$

The new Beltrami representation will then become as

$$X(x, y) \to (X_1 = x, X_2 = y, X_3 = P_R(x, y), X_4 = P_G(x, y), X_5 = P_B(x, y)) \tag{4.5}$$

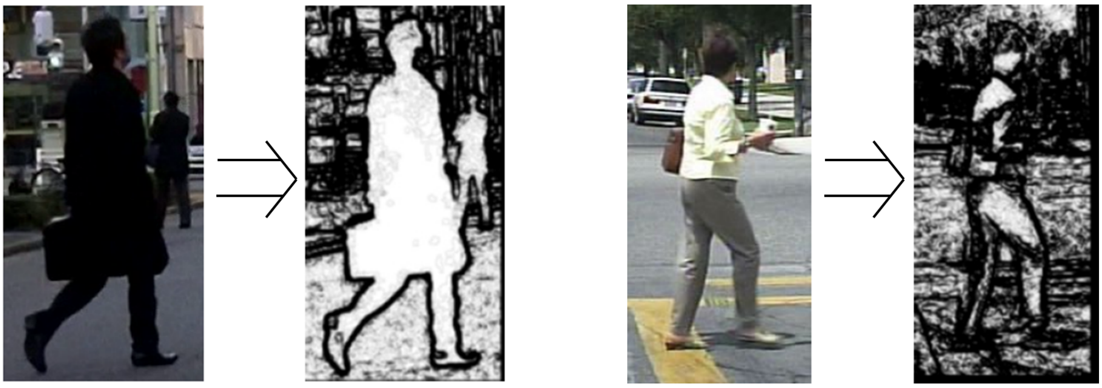
Calculating the metric tensor g_{xy} for this manifold, will yield

$$g_{xy} = \begin{pmatrix} 1 + \sum_{c \in \mathbb{C}} (\partial_x P_c(x, y))^2 & \sum_{c \in \mathbb{C}} \partial_x P_c(x, y) \partial_y P_c(x, y) \\ \sum_{c \in \mathbb{C}} \partial_x P_c(x, y) \partial_y P_c(x, y) & 1 + \sum_{c \in \mathbb{C}} (\partial_y P_c(x, y))^2 \end{pmatrix} \tag{4.6}$$

In the above equation, $\mathbb{C} = \{R, G, B\}$. Using this, the texture feature can be calculated as

$$T = \exp\Big(-\frac{\det(g_{xy})}{\sigma^2}\Big) \tag{4.7}$$

Here the Gaussian kernel acts as a low-pass filter which allows us to control the degree of details in the calculated feature by changing the value of the scaling parameter $\sigma > 0$.



(a) Subject with a single textured area

(b) Subject with multiple textured

areas

FIGURE 4.4: Example of texture feature for an input image.

Result of calculating this feature for an image is presented in Figure 4.4. In Figure 4.4(a), a subject is presented with an almost uniform color and texture in which, when the texture feature is calculated, the whole body will become as one block (single texture), while in (b), the subject has multiple textured regions as presented in the calculated texture map. As depicted here, this can be helpful in some cases while it can be very useful in other cases.

This texture map is then converted into a binary seed map by thresholding it. Since using one constant global threshold value would sometimes connect some parts of image with different textures to each other, due to not very distinct region boundaries, a local thresholding scheme is applied to have as much details as possible.

Write about local thresholding scheme The image edges are also removed from the texture map to make sure that different regions are separated from each other to the best extent as possible. For this, the threshold value is obtained in an $n \times n$ [pixels] window by calculating a weighted average inside that window.

4.2.4 Super-pixels Selection and Refinement Using Human Probability Map

Now that all image pixels are available as super-pixels/blocks, those that are related to the object-of-interest (the human body in this work). For this, the human probability map which shows where in the image frame most probably human body parts are, is used. Still, even with the best type of probability map, there is no guaranty that the probability map would match the current image (actually it most probably will not match exactly), a criterion needs to be set on how super-pixel/blocks are selected based on the probability map. For this, the following steps are prepared.

4.2.4.1 Selection

Assume that there are m super-pixels $\mathbb{S} = \{S_1, \ldots, S_m\}$ in the image. In the first stage, the super-pixel/blocks that have probability of more than a threshold P_{h_1} would be added to the set that can make the foreground mask as

$$\mathbb{M}_{\mathrm{PF}} = \{S_i | ^{\exists}(x, y) \in S_i; P(x, y) \geqslant P_{\mathrm{h}_1}\} \tag{4.8}$$

From these, the ones that overlap with the human probability body boundary more than P_{sp_1} in the probability map would be selected as the main part of a human body and put in $\mathbb{M}_{F_1}$ for creating the foreground mask. The rest of the super-pixels will be put in an auxiliary set $\mathbb{M}'_{\mathrm{PF}}$ for further process. Accordingly,

$$\mathbb{M}_{F_1} = \left\{S_i | \mathrm{score}(S_i) \geqslant P_{\mathrm{sp}_1}; S_i \subset \mathbb{M}_{\mathrm{PF}}\right\} \tag{4.9}$$

$$\mathbb{M}'_{\mathrm{PF}} = \left\{S_i | \mathrm{score}(S_i) < P_{\mathrm{sp}_1}; S_i \subset \mathbb{M}_{\mathrm{PF}}\right\} \tag{4.10}$$

where score(S_i) is

$$\mathrm{score}(S_i) = \frac{\#\{(x, y) | (x, y) \in S_i; P(x, y) \geqslant P_{h_1}\}}{\#\{S_i\}} \tag{4.11}$$

Here, #{•} represents the number of pixels included in the set. Other super-pixels with lower probability or the ones that contain human body parts would be collected in $\mathbb{M}'_{\mathrm{PF}}$ for further processing in the second step.

4.2.4.2 Refinement

In the second step, the super-pixels in $\mathbb{M}'_{\mathrm{PF}} = \{S'_1, \dots, S'_q\}$ are split into smaller parts. This time, the texture feature is extracted locally on these parts to achieve more details and feed the result to Watershed. Then, the new pieces are checked against the probability map to check if there are some parts with probability more than P_{h_2} as

$$\mathbb{M}_{F_A} = \{S'_i|^{\exists}(x, y) \in S'_i; P(x, y) \geqslant P_{\mathrm{h}_2}\} \tag{4.12}$$

From these blocks, the ones that overlap with the human probability body boundary more than P_{sp_2} in the probability map would be selected as the main part of a human body as

$$\mathbb{M}_{F_2} = \left\{S'_i|\mathrm{score}(S'_i) \geqslant P_{\mathrm{sp}_2}; S'_i \subset \mathbb{M}_{F_A}\right\} \tag{4.13}$$

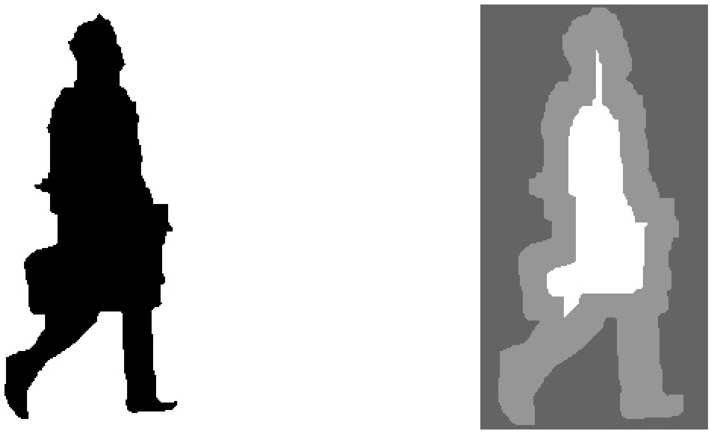
The foreground mask is then created by combining the super-pixels selected in the first step and the refined ones from the second step as

$$\mathbb{M}_F = \mathbb{M}_{F_1} \cup \mathbb{M}_{F_2} \tag{4.14}$$

This set is the final result for the rough shape of the human subject which will be further processed in the pixels-wise refinement step.

4.2.5 Further Refinement

After selecting related super-pixels to the best possible degree, a pixel-wise refinement is performed by using the Grab-cut method.



(a) Original mask

(b) Generated tri-map
(light gray: Foreground,
gray: Probably foreground,
dark gray: Probably background)

FIGURE 4.5: Example of a tri-map.

Here, the Grab-cut is used in the same way as in Chapter 3. Grab-cut is used here to make sure that at least some of the related parts that have not been selected in the super-pixel selection step can be segmented and presented as the final segmentation.

For using the Grab-cut at this step, first a binary mask is created by combining the super-pixels available in $\mathbb{M}_F$. This would be the base labeling map input to the Grab-cut segmentation process. For that, the mask must be first converted to a tri-map in the same manner as described in Chapter 3. An example of a tri-map is presented in Figure 4.5. After applying Grab-cut, a more refined segmentation result is obtained, which also includes some of the parts that have not been selected in the super-pixel selection step.

4.3 Experiments

4.3.1 Data-sets

Three different data-sets were prepared to test the proposed method and comparative methods.

TABLE 4.1: Parameters and their values in the proposed method.

	Parameter	**Value**
Super-pixel generation	P_{sp_1}	0.5
	P_{sp_2}	0.5
	P_{h_1}	0.4
	P_{h_2}	0.7
Local thresholding	n	6

The first two datasets are the same as the ones mentioned in 3.3.1. The private dataset with 180 images and the Caltech dataset with 100 images.

The third data-set is a subset created from the PennFudan data-set [123] which is a data-set created by both Pennsylvania and Fudan Universities. This set consists of 230 human subjects with different sizes. The images were taken with stationary cameras in different places. Usually the background in the images is complex, and in some cases, the color similarity between foreground and background is high. All images in the dataset are taken in day time, RGB color images, and contain a single subject.

As for the ground-truth for evaluating the segmentation accuracy, for the first and second data-sets, ground-truth has been created by manually segmenting the human subjects and turning the results into a binary image. In case of the PennFudan data-set, the ground truth is provided for each human subject by its distributor.

4.3.2 Setup

As it has been mentioned before in 4.1, for finding human body parts, a probability map is used. This map provides us with an estimation of the existence of human body parts in different places of a selected window. This probability map is generated off-line.

add description for Tabel 4.1

4.3.3 Comparative Methods

To test the validity of the proposed method, it was compared with some automatic and interactive segmentation methods.

As for the automatic methods, first Grab-cut [19] which is famous due to simple interaction and iterative energy minimization, was prepared. It has been made automatic by providing the system with the bounding box given to it. The second method was SSFSeg proposed in Chapter 3 which uses a human shape model alongside Grab-cut for automatic segmentation.

As for the interactive segmentation methods for comparison, the first one was Watershed algorithm [70] which can be considered a conventional method. It has fast response time and tends to give coherent segmentation regions. The others were Efficient Graph-cut segmentation [68], Planarcut [67], Onecut [66], and Convexity shape prior [69]. The main reason for this selection is that aside from Watershed algorithm [70] and Convexity shape prior [69], other methods have tried a different aspect for solving the graph-based segmentation problem. Each method was briefly explained in Chapter 1. The two latter methods are new methods which have showed accurate segmentation results.

4.3.4 Results

Some experiments have been performed to validate the proposed system. The results have been compared to the methods mentioned in 4.3.3. For comparison, the accuracy was calculated based on the following formula:

$$\text{Accuracy } [\%] = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \times 100 \tag{4.15}$$

Here, "TP" represents True Positive which is the number of pixels that are correctly selected as foreground in an image, "TN" represents True Negative which is the number of pixels that are correctly selected as background in an image, "FP" represents False

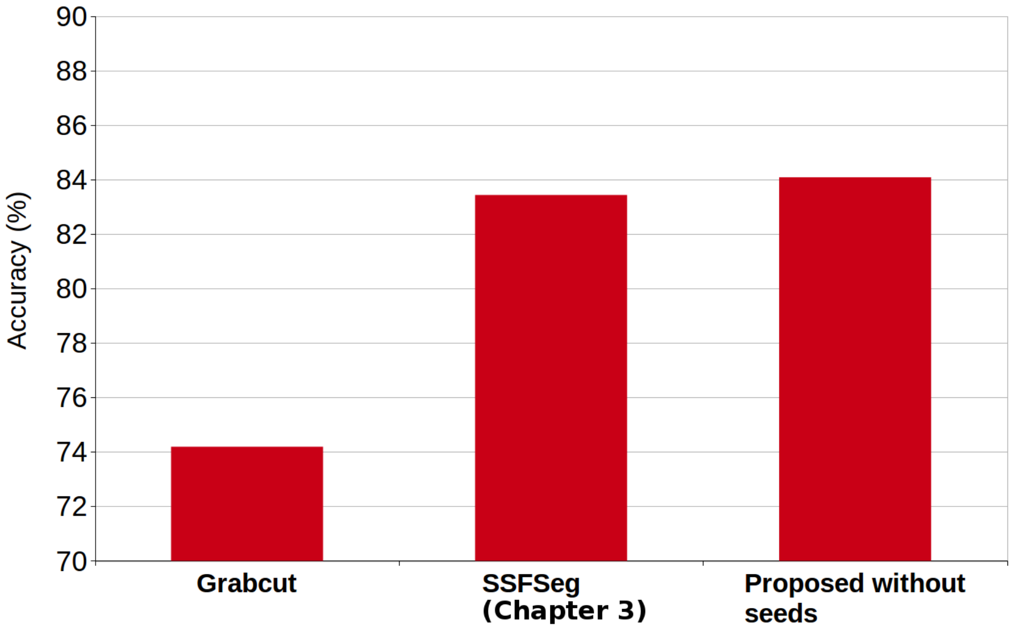


Figure 4.6: Automatic segmentation: Comparison of average segmentation accuracy between comparative methods and the proposed method which uses human probability map and super-pixels.

Positive which is the number of pixels that are wrongly selected as foreground in an image, and "FN" represents False Negative which is the number of pixels that are selected as background in an image by mistake.

4.3.4.1 Automatic Segmentation

If the proposed method is used for automatic segmentation, the results of the system which uses human probability map and super-pixels become as presented in Figure 4.6. Comparison is done between the proposed method, the automated Grab-cut [19], and SSFSeg (Chapter 3). As in the graph of Figure 4.6, by combining the human probability map and super-pixels, the accuracy becomes significantly higher than the Grab-cut while it wins against the SSFSeg method.

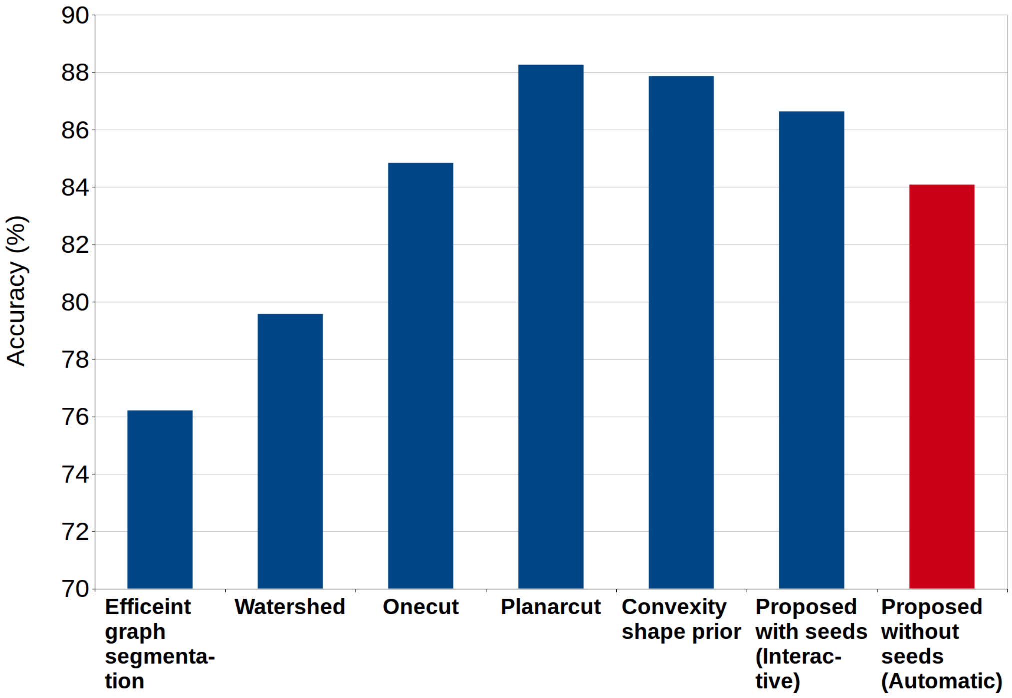
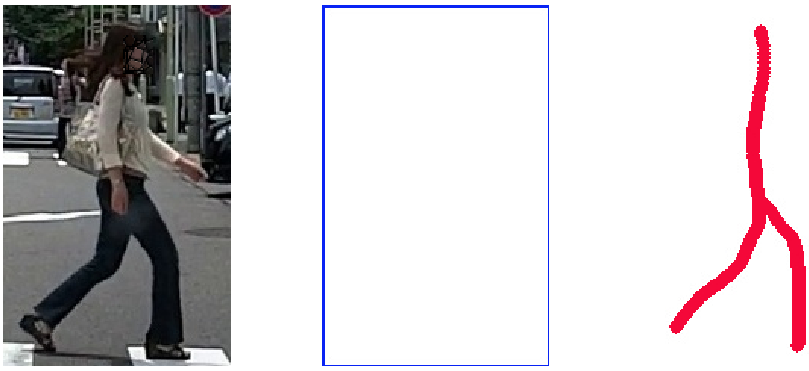


FIGURE 4.7: Interactive segmentation: Comparison of average segmentation accuracy between different segmentation methods and the proposed method which uses human probability map and super-pixels.

4.3.4.2 Interactive Segmentation

If the proposed method is used for interactive segmentation, results of the system which uses human probability map and super-pixels become as presented in Figure 4.7. Here, the segmentation accuracy of the system is compared with some comparative interactive methods. It can be seen that although the proposed uses a relatively simpler way of doing things, the segmentation results are comparable with the state-of-the-art interactive segmentation methods. Note that for interactive segmentation, some foreground and background seeds are necessary. For this, all of the methods were provided with the same type of seeds. As presented in Figure 4.8, for background seeds, a rectangle around the picture was provided, while for the foreground, some seeds were selected manually so that they cover the basic skeletal shape of human body and almost cover the subject for fair judgment.



(a) Input image (b) Background seeds (c) Foreground seeds

Figure 4.8: Manual seed selection.

4.4 Discussion

Here, a system that can automatically segment human subject from an image was proposed. By converting the image into a puzzle, using a relatively simple texture feature and human probability map, it was shown that good segmentation results could be achieved. Using the proposed method not only solves the main problem of Graph-cut based methods which uses just a color distance feature for distinction between two regions, but also allows the segmentation of the human subject automatically with more accuracy. Compared to the original Grab-cut, the accuracy improvement is significant while it is increased even compared to the method proposed in Chapter 3 as depicted in Figure 4.6.

Although the proposed method is automatic by nature, it can also be used as an interactive segmentation method. The result of using the system in interactive mode compared to the same type of comparative methods is presented in Figure 4.7. As it can be seen, even if the final refinement stage uses the Grab-cut method, the accuracy of the system in interactive mode is almost on par with recently proposed interactive methods, while in automatic mode, the result becomes comparable with state-of-the-art methods and much better than the conventional ones. If instead of Grab-cut, another method is used for refinement, the accuracy might improve further. Some examples of segmentation quality in comparison with other methods are depicted in Figs. 4.9 and 4.10. Comparison of all methods is also presented in Tables 4.2 and 4.3.

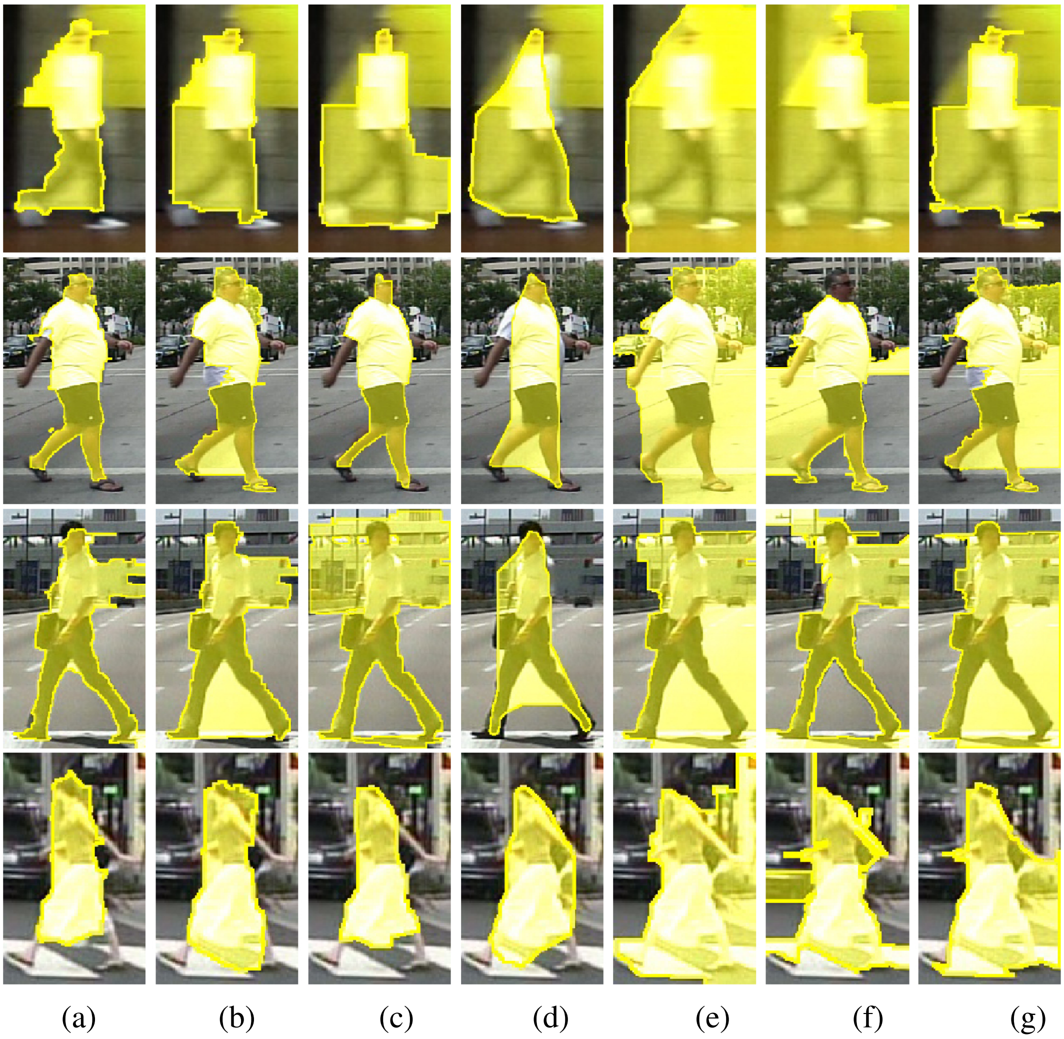


FIGURE 4.9: Good segmentation examples from the Caltech data-set: (a) Proposed method, (b) Planarcut [67], (c) Onecut [66], (d) Convexity shape prior [69], (e) Grab-cut [19], (f) Efficient Graph-cut segmentation [68], and (g) Watershed [70].

Still, since here a simple probability map and texture feature are used, in some cases, the desired segmentation result is not obtained. An example of this is presented in Figure 4.11. The main reason of failure for Figure 4.11 (a) is the wrong probability prediction by the human probability map. The reason for Figure 4.11 (b) is the miscalculation in the texture because of the similarity between the color and texture of the foreground object and a part of the background which leads to the creation of a super-block. When this super-block is checked against P_{sp_1}, it will be considered as part of the background.

The process of creating the probability map was introduced in 4.2.1. It is also good to mention that the off-line procedure of creating the map can be accompanied with an on-line updating scheme to improve the map by adding the mask of the segmented subjects.

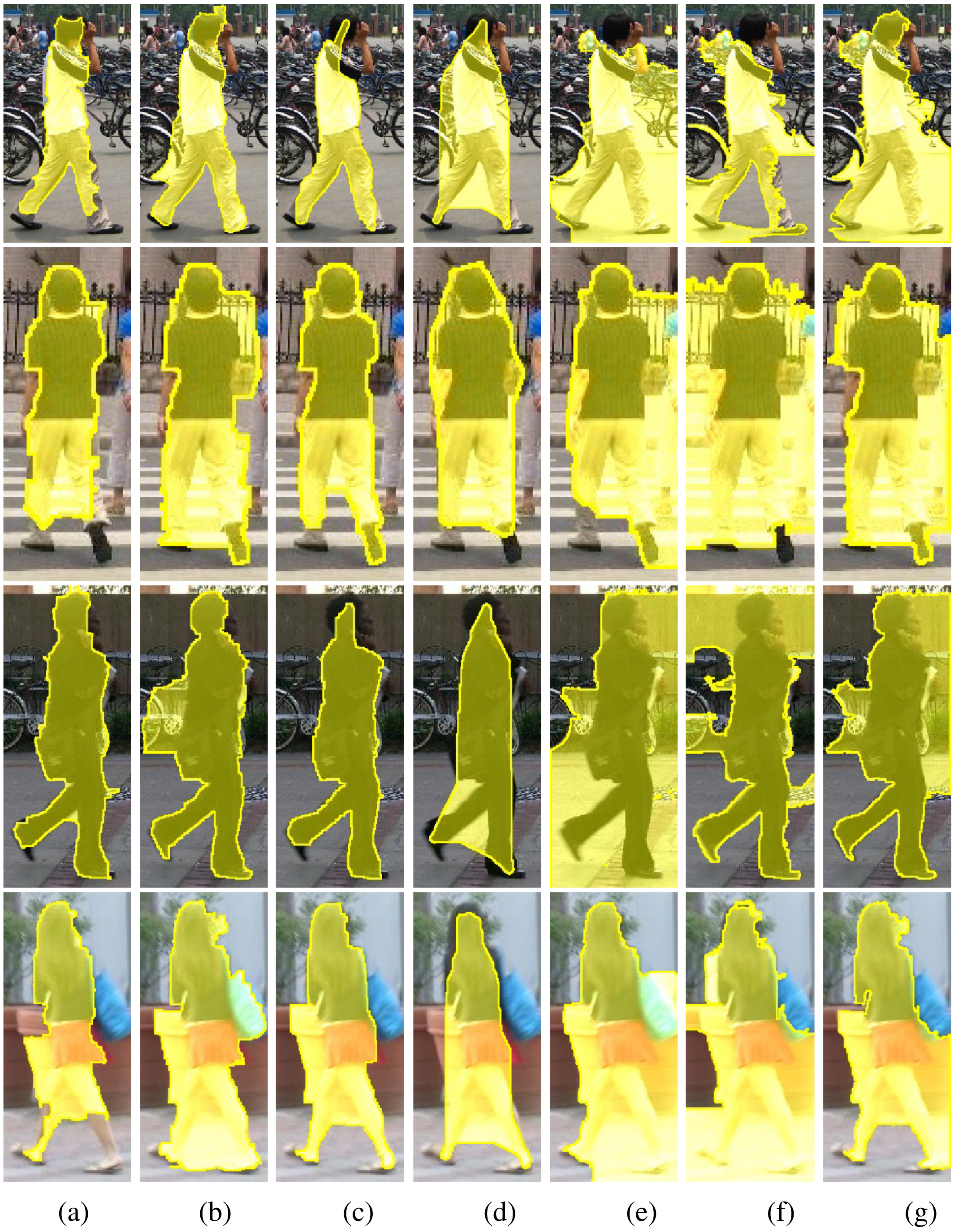


FIGURE 4.10: Good segmentation from PennFudan data-set: (a) Proposed method, (b) Planarcut [67], (c) Onecut [66], (d) Convexity shape prior [69], (e) Grab-cut [19], (f) Efficient Graph-cut segmentation [68], and (g) Watershed [70].

Although this is a possible way for improvement, the proposed method only incorporates the off-line stage. Also, it is good to note that even though different methods can be used to turn an image into super-pixels, the Watershed had satisfactory results for the purpose based on the provided texture seeds. As depicted in Figure 4.12, by just looking at the super-pixels, the boundary of human body is recognizable. Still, using

Table 4.2: Performance comparison between different methods in interactive mode.

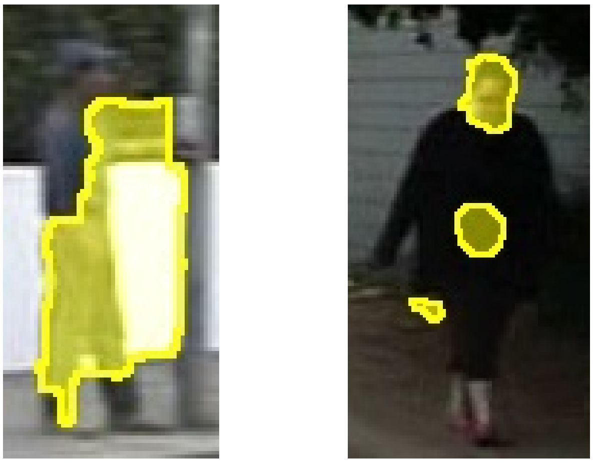
Method	**Accuracy (%)**		
	Private	**Caltech**	**PennFudan**
Efficeint graph segmentation [68]	77.07	73.29	78.27
Watershed [70]	79.23	78.71	80.77
Onecut [66]	83.86	84.92	85.74
Planarcut [67]	89.72	87.65	87.42
Convexity shape prior [69]	88.90	88.12	86.58
Proposed with seeds	89.31	86.12	84.48

Table 4.3: Performance comparison between different methods in automatic mode.

Method	**Accuracy (%)**		
	Private	**Caltech**	**PennFudan**
Proposed without seeds	87.36	85.64	79.24
SSFSeg (Chapter 3)	83.66	86.03	80.60
Grab-cut [19]	72.08	71.45	79.03

other methods might improve the system results.

Although the proposed method performs the segmentation of the image in one iteration in contrast to the method proposed in Chapter 3, the segmentation result is almost the same on Caltech and PenFudann datasets while improving by 4% on the private dataset. The main reason for the improvement is the use of probability map and super-pixels instead of initializing a shape model on the image that might not match completely. Also the computation time has been significantly reduced. For example, the proposed method can segment all 180 images of the private dataset in 184.8 (1.03 seconds per image) while the Grab-cut performs the task in 39.6 seconds (0.22 seconds per image), and SSFSeg performs it in 3.35 hours (67 seconds per image).



(a) Failure due to probability map (b) Failure due to texture map

FIGURE 4.11: Two examples of segmentation failure from Caltech data-set.

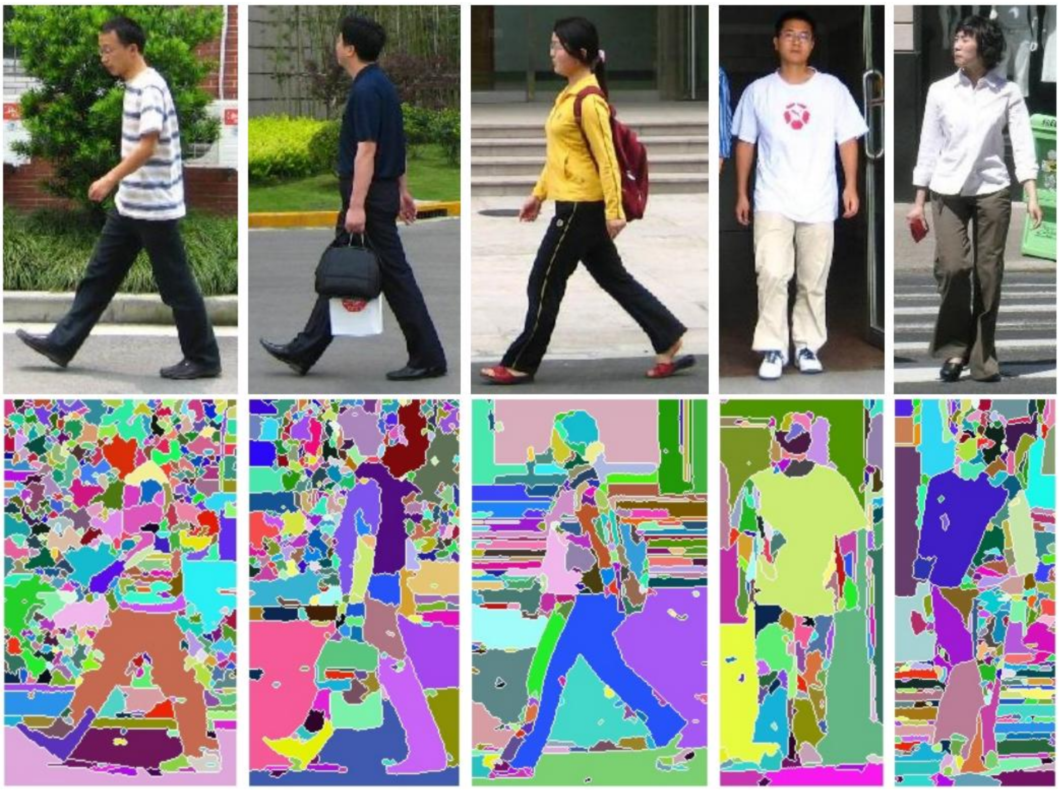


FIGURE 4.12: Watershed texture based super-pixels generation results (Input images from PennFudan dataset [3]).

4.5 Summary

In this chapter, an automatic method for human subject segmentation in single shot frames was presented with good accuracy comparable to some state-of-the-art methods. The main idea was to segment an image into multiple super-pixels and then try to find those which were related to human body parts using a human body probability map. Although the used map is relatively simple, the system showed promising results in segmentation. Also, even though the system is automatic, we have confirmed the possibility of using it interactively.

The proposed method in this chapter not only solves the color/texture variation problem mentioned in Chapter 1, it also improves the method proposed in Chapter 3 considerably, in terms of computation time. Although the method proposed here and in Chapter 3 take different approaches to solve the human segmentation problem, both are common in the concept that using various kinds of prior information about the object to-be-segmented will considerably affect the final segmentation result. Again, the same as its previously proposed counterpart, this method is not limited to single-frame image and can be easily adapted to multi-frame scenario. Also the application is not just limited to the human subjects, but if available, the system can be trained to segment other types of objects easily.

That said, there are some problems that have to be considered for further improving the method proposed in this chapter. The following points are the most important ones.

1. The probability map used in this chapter is a very simple one. It would be more effective to use other types of probability maps which can provide more accurate information for the human body.

2. The refinement stages and the Grab-cut improvement in complex images sometimes become insignificant which implies that using the methods which do not solely rely on color features might be more useful. It is still good to note that, even with these problems, the proposed method shows a significant improvement compared to the original Grab-cut algorithm and is also automatic.

3. Optimizing the code would also reduce the time needed for segmentation giving more possibilities to the applications this method can be used for.

4. Since the utilized probability map is simple, the system cannot cope with major occlusion which means in such cases, the segmentation results will most probably be not satisfactory.

Chapter 5

Conclusion

5.1 Summary

In this thesis, the problem of human subject segmentation using automatic graph-based segmentation has been addressed. As mentioned in Chapter 1, two main problems for human subject segmentation are the variation in the shape of the body due to articulation, and the variation in the color/texture of the body due to numerous combinations of color/texture in the clothing of humans.

As mentioned in Chapter 1, the following problems were the main focus in this thesis.

1. Shape variations due to human body movements
2. Shape variations due to human wearing different clothes
3. Variation in color and texture of the clothing
4. Complexity of the scene

The first problem relates to many variations in the shape of human body. Especially when the subject moves or the video is recorded from different views and angles. This makes the modeling task difficult and a model that can cope with all of the variations can be very complex.

The second problem comes from the fact that humans wear various types of cloths in different situations. Combination between the movement and change in the shape caused by a human subject wearing different types of clothing which can affect their general shape (Rain coats, coats, T-shirts, normal shirts, and so on), adds to the complexity of the model.

The third problem about cloths is the color and texture of the cloth, since an unimaginable combinations of cloths is available. These combinations not only change between different people but even for one person they will change in different occasions. The clothing is also different between genders, so for example, a model designed specifically for men might not work for women.

The forth problem which makes the task harder is the complexity of the scene that a human subject is being recorded in. Even the simplest real-world cases for human eyes has proved to be quite challenging for the computer to understand and differentiate. This means that finding a model for the background will also become difficult. Especially if a moving camera is used, the problem will ascend to another level of difficulty since now a dynamic model to cope with the situation is necessary.

To solve the above mentioned problems, two methods were proposed in this thesis. Those methods tried to make use of the Grab-cut [19] framework which as mentioned in Chapter 1 is based on Graph-cut frame work. The main reason for this choice is that, the Grab-cut has the capability to incorporate different kinds of prior informations in its framework as mentioned in Chapter 1 and presented in Table 5.1. As a result, the proposed methods not only improved the original method in terms of accuracy, they also relieved the user from need to interact with the systems and make corrections.

The main contributions of this thesis can be explained in the following points:

- Making a coarse-to-fine feedback framework that can be applied to interactive methods similar to the Grab-cut framework, and turn them into unsupervised methods when the time consumption is of no concern.
- Implementing the same idea by utilizing the human probability map and superpixels for application with need for less time consumption.

Table 5.1: Comparison between different spatially guided object segmentation methods. + means the feature is supported, −/+ means it is possible to support the feature with additional implementation, and − means the feature is not supported.

	Region-based			Energy-based					
Methods	Region growing	Region merging	Hybrid	Active contours	Mumford-Shah	Baysian	Graph-cut	Other graph-based	Watershed
Automatic initialization	−	−	−	−/+	−/+	−/+	−/+	−/+	+
Various prior information inclusion	−/+	−/+	−/+	−/+	−/+	+	+	+	−/+
Complex object	−/+	−/+	−/+	−	−	−	+	+	+
Computation time	+	+	+	−	−/+	−/+	+	−/+	+
Robust against noise	−	−	−	−/+	−/+	+	+	+	−
Robust against weak boundaries	−	−	−	−/+	−/+	+	+	+	−
Multiple object segmentation	−/+	−/+	−/+	−	+	+	+	+	−/+

- Incorporating a simplistic texture feature information to the segmentation process which in combination with super-pixels, human probability map, and Grab-cut lead to an accurate automatic segmentation system.

To summarize the work presented in this thesis, in Chapter 3, for overcoming the changes in the shape and color, a method using the conventional Statistical Shape Models (SSM) and Grab-cut segmentation algorithm was proposed. SSM uses the statistic framework to model changes in the shape of an object in Eigen-space and by analyzing it and finding the most dominant changes, it can even try to generate some new shapes with the properties of the original one, while the Grab-cut segmentation algorithm tries to learn image color distributions and segment the foreground object based on that in an Markov Random Fields (MRF) framework. By connecting these two methods through a feedback system, it becomes possible to propose a coarse-to-fine scheme for model generation and refinement which in combination with Grab-cut, leads to accurate segmentation results for human subjects.

Although using the coarse-to-fine scheme proved to be very effective, there were some problems that needed to be solved:

1. Even if the window containing the human subject is provided, the exact location of the body is unknown. As a result, initializing the system with a proper mask and in correct location, becomes difficult.
2. Even if the generated masks are detailed on the shape of the body, there is no guarantee that they will match the actual human body shape completely. Although the Grab-cut segmentation stage solves this problem to a good extent, it might not be able to cope with the problem in some cases.
3. Coarse-to-fine scheme has a recursive nature. This means that the process of shape model refinement must be performed iteratively, which as mentioned in 4.4 will be taxing on computational power and time.
4. Since the segmentation step of the proposed method uses the Grab-cut framework in which just color feature is used in cases where there is a change in texture or color of one object as in Figure 4.1, the system sometimes fail to produce a correct segmentation.

To cast aside the above mentioned problems in Chapter 4, a system which makes use of a human probability map, super-pixels, and Grab-cut framework was proposed. Using the probability map will solve the two first problems mentioned above as now we have an estimation about the location and general pose of the body (although very simple), and using a texture based super-pixel generation gives us the advantage of using texture feature in the framework. As there is no need for iterative shape model refinement, the system also becomes much faster than before. The main idea comes from jigsaw puzzle; If the image is divided into regions based on their color/texture, each part of the human body then, becomes like a piece of the puzzle. So, an image can be considered as a puzzle with multiple pieces in which the human body occupies some of them. By selecting the right pieces, a somewhat rough (or even fine) shape of the body is obtained and by using Grab-cut, the human subject is segmented accurately. Following this idea, not only the system becomes automatic, but also the accuracy of the system is improved. Also, just by using the information of a single image, it is possible to achieve segmentation results with accuracy comparable to the state-of-the-art and much better than conventional methods while having a relatively simpler model. It is also good to note that this method can be used in both automatic and interactive segmentation manners.

In this thesis, the problem of human subject segmentation has been considered. The result is two automatic full human body segmentation methods in still images which take advantage of the accuracy of interactive segmentation methods (Grab-cut in this thesis) in an unsupervised manner. Although the proposed methods are mainly focused on still image segmentation, nothing prevents them not to work on videos. Still, some improvement is needed from them to become practical for real-time applications.

5.2 Future Works

The main goal of this thesis is to propose methods for automatic human subject segmentation. Since as mentioned in the introduction of this thesis, human subject segmentation has application in many fields. Different applications can be imagined for a this kind of methods. From them, the following applications are note worthy.

- Virtual reality systems (also entertainment systems) - Generating more realistic simulations.
- Driver assistance systems - Detecting the pedestrians and their intention and assisting the driver for safe driving.
- Automatic navigations systems - The same as above but for accident prevention and so on.
- Video archiving
- Surveillance - Detecting intruders.

Still, one of the best application fields for the proposed methods can be the automotive industry, especially in nowadays, intelligent vehicles. As now, automatic driving systems which were part of Science Fiction novels and movies until a while ago, is becoming realistic, it can be used for assisting a human driver or directly be used in an automatic navigation system as part of a driver safety system. Although, the methods proposed in this thesis are not currently working in real time, looking on how fast is the advances in the world of computers (for example Intel's Xeon Phi series with up to 61 cores and 1.2 TFLOPS performance are available now but expensive for normal consumer use), it will be possible in few years later.

Still for the proposed methods to become completely practical for the mentioned applications some points must be considered for further improvement. One of those cases is to make the model capable of coping with occlusion as it has not been considered in this thesis. Although the proposed methods can adapt themselves with small part occlusions, when there is a considerable amount of occlusion, either both objects (can be two humans occluding each other or an object and a human), or none of them will be segmented with the current systems.

Another thing to consider is the probability map used for the proposed method in Chapter 4. As it is also mentioned there, the utilized model was a very simple one made by just adding some hand-segmented silhouettes together, for the sake of simplicity of the system. The system should make use of more sophisticated models for further improvement. Another problem to mention is that, although the proposed systems are theoretically capable of segmenting human subjects from images taken from angular view (like from a surveillance camera), it has not been practically tested because of the lack of samples in the used datasets. So, the application to datasets with higher varieties of samples including angular view needs to be considered.

To summarize, to further improve the work in this thesis, the following points should be considered.

- Make a more complete training dataset for the SSM generation step which includes more variations in the model (for the method proposed in Chapter 3).

- Optimize the code, so the system becomes capable of real-time segmentation.
- Extend the algorithm and devise a multi-frame segmentation scheme.
- Find a better method to generate and use the probability map.
- Use better texture descriptors.
- Unify multiple stages in one framework.
- Make use of other frameworks or methods like Planarcut [67], in the segmentation stage.

The proposed feedback and refinement framework in Chapter 3 is relatively simple and easy to train and implement which makes it very useful for turning other types of interactive methods similar to Grab-cut to an unsupervised type. Since the feedback-refinement framework provides the segmentation procedure with the information about the shape of the object to-be-segmented and refine it iteratively, it is possible to replace the user inputs with this automatic process. Also, although SSM is used to model shape variations of human body, the application is not limited to just human segmentation. It is completely possible to adapt the framework for other types of objects (preferably the ones with distinct shape variations) by training the SSM with the desired object and use it for the segmentation as a part of the proposed method in Chapter 3 or just use the feedback-refinement framework alongside another method.

The same can be done for the framework introduced in Chapter 4. By creating a probability map of the desired object, again it is possible to use the super-pixel generation, selection, and refinement process of the proposed method alongside another methods (again interactive methods like Grab-cut) or use the whole proposed method for the object-of-interest segmentation. Still, it is good to note that it is advisable to make a more accurate probability map by using more advanced methods for achieving better results. For example, it might be possible to utilize the concept of Deep Learning which is based on the Neural Networks for categorization and classification, and has become very active area recently, to train on much more samples and create a more accurate map. Also, some methods like Convolutional Neural Networks (CNN), Deep

Convolutional Neural Networks, or Deep Neural Decision Forests (DNDF) [124] which are trying to make the task of learning and classification at the same time, might be useful too. Since these methods try to learn from input training samples on themselves, they make the task of learning and classification very easy. But the tradeoff is the problem of tuning the network for the best performance.

Bibliography

[1] L. G. Shapiro and G. C. Stockman, *"Image segmentation", Computer Vision*. New Jersey: Prentice-Hall, 2001.

[2] M. Agarwal and V. Singh, "A methodological survey and proposed algorithm on image segmentation using genetic algorithm," *Int. J. of Computer Applications*, vol. 67, pp. 7–17, 2013.

[3] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Proc. 2009 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, (Miami, FL, USA), pp. 304–311, Jun. 2009.

[4] S. R. Vantaram and E. Saber, "Survey of contemporary trends in color image segmentation," *J. of Electronic Imaging*, vol. 21, no. 4, pp. 040901-1–040901-28, 2012.

[5] H.-D. Cheng, X. Jiang, Y. Sun, and J. Wang, "Color image segmentation: Advances and prospects," *Pattern Recognition*, vol. 34, no. 12, pp. 2259–2281, 2001.

[6] M. Sridevi and C. Mala, "A survey on monochrome image segmentation methods," *Procedia Technology*, vol. 6, pp. 548–555, 2012.

[7] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary information integration," in *Proc. 2002 European Conf. on Computer Vision (ECCV)*, (Antibes, France), pp. 408–422, May 2002.

[8] R. M. Haralick and L. G. Shapiro, “Glossary of computer vision terms,” *Pattern Recognition*, vol. 24, no. 1, pp. 69–93, 1991.

[9] M. Prantl, H. Ganster, and A. Pinz, “Glossary of computer vision terms in connection to information fusion,” *Vision Milestones*, pp. 149–159, 1995.

[10] T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE Trans. on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[11] C. Xu and J. L. Prince, “Snakes, shapes, and gradient vector flow,” *IEEE Trans. on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[12] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” in *Proc. 1995 IEEE Int. Conf. on Computer Vision (ICCV)*, (Boston, MA, USA), pp. 694–699, Jun. 1995.

[13] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *Int. J. of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[14] D. Zhang, O. Javed, and M. Shah, “Video object segmentation through spatially accurate and temporally dense extraction of primary object regions,” in *Proc. 2013 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, (Portland, OR, USA), pp. 628–635, Jun. 2013.

[15] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *Int. J. of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[16] V. Gulshan, V. Lempitsky, and A. Zisserman, “Humanising grabcut: Learning to segment humans using the Kinect,” in *Proc. 2011 IEEE Int. Conf. on Computer Vision (ICCV) Workshops*, (Barcelona, Spain), pp. 1127–1133, Nov. 2011.

[17] M. P. Kumar, P. Ton, and A. Zisserman, “Obj cut,” in *Proc. 2005 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, (San Diego, CA, USA), pp. 18–25, Jun. 2005.

[18] Y. Boykov and M. P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in ND images,” in *Proc. 2001 IEEE Int. Conf. on*

Computer Vision (ICCV), vol. 1, (Vancouver, BC, Canada), pp. 105–112, Jul. 2001.

[19] C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: Interactive foreground extraction using iterated graph cuts,” *ACM Trans. on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[20] B. Peng and O. Veksler, “Parameter selection for graph cut based image segmentation.,” in *Proc. 2008 British Machine Vision Conf.*, (Leeds, UK), Sep. 2008.

[21] M. Szummer, P. Kohli, and D. Hoiem, “Learning CRFs using graph cuts,” in *Proc. 2008 European Conf. on Computer Vision (ECCV)*, (Marseille, France), pp. 582–595, Oct. 2008.

[22] Z. Kuang, D. Schnieders, H. Zhou, K.-Y. Wong, Y. Yu, and B. Peng, “Learning image-specific parameters for interactive segmentation,” in *Proc. 2012 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, (Providence, RI, USA), pp. 590–597, Jun. 2012.

[23] S. Prakash, S. Das, and R. Abhilash, “Snakecut: An integrated approach based on active contour and GrabCut for automatic foreground object segmentation,” *Electronic Letters on Computer Vision and Image Analysis*, vol. 6, no. 3, pp. 13–28, 2007.

[24] Y. Li, J. Sun, and H.-Y. Shum, “Video object cut and paste,” *ACM Trans. on Graphics*, vol. 24, no. 3, pp. 595–600, 2005.

[25] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, “Lazy snapping,” *ACM Trans. on Graphics*, vol. 23, no. 3, pp. 303–308, 2004.

[26] E. N. Mortensen and W. A. Barrett, “Intelligent scissors for image composition,” in *Proc. 1995 ACM Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, (Los Angeles, CA, USA), pp. 191–198, Aug. 1995.

[27] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof, “TVSeg—Interactive total variation based image segmentation.,” in *Proc. 2008 British Machine Vision Conf.*, (Leeds, UK), Sep. 2008.

[28] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, “Geodesic star convexity for interactive image segmentation,” in *Proc. 2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, (San Francisco, CA, USA), pp. 3129–3136, May 2010.

[29] T. Gandhi and M. M. Trivedi, “Pedestrian collision avoidance systems: A survey of computer vision based recent studies,” in *Proc. 2006 IEEE Conf. on Intelligent Transportation Systems*, (Toronto, ON, Canada), pp. 976–981, Sep. 2006.

[30] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[31] W. Tao, H. Jin, and Y. Zhang, “Color image segmentation based on mean shift and normalized cuts,” *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 37, no. 5, pp. 1382–1389, 2007.

[32] T. Kohonen, “The self-organizing map,” *Proc. of IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[33] S. Krinidis and V. Chatzis, “A robust fuzzy local information c-means clustering algorithm,” *IEEE Trans. on Image Processing*, vol. 19, no. 5, pp. 1328–1337, 2010.

[34] K. S. Tan and N. A. M. Isa, “Color image segmentation using histogram thresholding—Fuzzy C-means hybrid approach,” *Pattern Recognition*, vol. 44, no. 1, pp. 1–15, 2011.

[35] S. Bhattacharyya, U. Maulik, and P. Dutta, “Multilevel image segmentation with adaptive image context based thresholding,” *Applied Soft Computing*, vol. 11, no. 1, pp. 946–962, 2011.

[36] R. Adams and L. Bischof, “Seeded region growing,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[37] L. Chen, H. Lin, and S. Li, “Depth image enhancement for Kinect using region growing and bilateral filter,” in *Proc. 2012 Int. Conf. on Pattern Recognition (ICPR)*, (Tsukuba, Japan), pp. 3070–3073, Nov. 2012.

[38] J. Fan, D. K. Yau, A. K. Elmagarmid, and W. G. Aref, “Automatic image segmentation by integrating color-edge extraction and seeded region growing,” *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 1454–1466, 2001.

[39] T. Kong, G. Yang, and L. Yang, “A new finger-knuckle-print ROI extraction method based on probabilistic region growing algorithm,” *Int. J. of Machine Learning and Cybernetics*, vol. 5, no. 4, pp. 569–578, 2014.

[40] P. Yu, A. Qin, and D. Clausi, “Unsupervised polarimetric SAR image segmentation and classification using region growing with edge penalty,” *IEEE Trans. on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1302–1317, 2012.

[41] Y. Morimoto, H. Ishii, and S. Morishita, “Efficient construction of regression trees with range and region splitting,” *Machine Learning*, vol. 45, no. 3, pp. 235–259, 2001.

[42] R. Ohlander, K. Price, and D. R. Reddy, “Picture segmentation using a recursive region splitting method,” *Computer Graphics and Image Processing*, vol. 8, no. 3, pp. 313–333, 1978.

[43] K. Fu, C. Gong, Y. Yun, Y. Li, I. Y.-H. Gu, J. Yang, and J. Yu, “Adaptive multi-level region merging for salient object detection,” in *Proc. 2014 British Machine Vision Conf.*, (Nottingham, UK), Sep. 2014.

[44] J. Ning, D. Zhang, C. Wu, and F. Yue, “Automatic tongue image segmentation based on gradient vector flow and region merging,” *Neural Computing and Applications*, vol. 21, no. 8, pp. 1819–1826, 2012.

[45] B. Peng, L. Zhang, D. Zhang, and J. Yang, “Image segmentation by iterated region merging with localized graph cuts,” *Pattern Recognition*, vol. 44, no. 10, pp. 2527–2538, 2011.

[46] A. Kovacs and T. Sziranyi, "Harris function based active contour external force for image segmentation," *Pattern Recognition Letters*, vol. 33, no. 9, pp. 1180–1187, 2012.

[47] C. Xu and J. L. Prince, *"Gradient Vector Flow", Computer Vision—A Reference Guide*. New York: Springer, 2014.

[48] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, 1996.

[49] W. Gao, H. Ai, and S. Lao, "Adaptive contour features in oriented granular space for human detection and segmentation," in *Proc. 2009 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, (Miami, FL, USA), pp. 1786–1793, Jun. 2009.

[50] Q. Ge, L. Xiao, J. Zhang, and Z. H. Wei, "A robust patch-statistical active contour model for image segmentation," *Pattern Recognition Letters*, vol. 33, no. 12, pp. 1549–1557, 2012.

[51] N. Houhou, J.-P. Thiran, and X. Bresson, "Fast texture segmentation based on semi-local region descriptor and active contour," *Numerical Mathematics: Theory, Methods and Applications*, vol. 2, no. 4, pp. 445–468, 2009.

[52] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Proc. 2000 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, (Hilton Head Island, SC, USA), pp. 316–323, Jun. 2000.

[53] A. Dubrovina, G. Rosman, and R. Kimmel, *Active contours for multi-region image segmentation with a single level set function*. Berlin: Springer, 2013.

[54] N. Paragios and R. Deriche, "Coupled geodesic active regions for image segmentation: A level set approach," in *Proc. 2000 European Conf. on Computer Vision (ECCV)*, (Dublin, Ireland), pp. 224–240, Jun. 2000.

[55] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, "Fast geodesic active contours," *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 1467–1475, 2001.

[56] X. Cai, R. Chan, and T. Zeng, "A two-stage image segmentation method using a convex variant of the Mumford–Shah model and thresholding," *SIAM J. on Imaging Sciences*, vol. 6, no. 1, pp. 368–390, 2013.

[57] N. Y. El-Zehiry and L. Grady, "Combinatorial optimization of the discretized multiphase Mumford–Shah functional," *Int. J. of Computer Vision*, vol. 104, no. 3, pp. 270–285, 2013.

[58] I. Posirca, Y. Chen, and C. Z. Barcelos, "A new stochastic variational PDE model for soft Mumford–Shah segmentation," *J. of Mathematical Analysis and Applications*, vol. 384, no. 1, pp. 104–114, 2011.

[59] A. Vitti, "The Mumford–Shah variational model for image segmentation: An overview of the theory, implementation and use," *ISPRS J. of Photometry and Remote Sensing*, vol. 69, pp. 50–64, 2012.

[60] J. S. Borges, J. Bioucas-Dias, and A. R. Marcal, "Bayesian hyperspectral image segmentation with discriminative class learning," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2151–2164, 2011.

[61] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva, "A Bayesian-MRF approach for PRNU-based image forgery detection," *IEEE Trans. on Information Forensics and Security*, vol. 9, no. 4, pp. 554–567, 2014.

[62] R. Kulkarni, M. Tuller, W. Fink, and D. Wildenschild, "Three-dimensional multiphase segmentation of X-ray CT data of porous materials using a Bayesian Markov random field framework," *Vadose Zone J.*, vol. 11, no. 1, pp. 74–85, 2012.

[63] G. Veni, Z. Fu, S. P. Awate, and R. T. Whitaker, "Bayesian segmentation of atrium wall using globally-optimal graph cuts on 3D meshes," in *Proc. 2013 Int. Conf.*

on Information Processing in Medical Imaging (IPMI), (Asilomar, CA, USA), pp. 656–667, Jul. 2013.

[64] L. Zhang and Q. Ji, "Image segmentation with a unified graphical model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1406–1425, 2010.

[65] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[66] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov, "Grabcut in one cut," in *Proc. 2013 IEEE Int. Conf. on Computer Vision (ICCV)*, (Sydney, Australia), pp. 1769–1776, Dec. 2013.

[67] F. R. Schmidt, E. Toppe, and D. Cremers, "Efficient planar graph cuts with applications in computer vision," in *Proc. 2009 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, (Miami, FL, USA), pp. 351–356, Jun. 2009.

[68] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[69] L. Gorelick, O. Veksler, Y. Boykov, and C. Nieuwenhuis, "Convexity shape prior for segmentation," in *Proc. 2014 European Conf. on Computer Vision (ECCV)*, (Zürich, Germany), pp. 675–690, Sept. 2014.

[70] F. Meyer and S. Beucher, "Morphological segmentation," *J. of Visual Communication and Image Representation*, vol. 1, no. 1, pp. 21–46, 1990.

[71] R. Szeliski, *Computer Vision: Algorithms and Applications*. London, UK: Springer, 2011.

[72] H. Gao, W. Lin, P. Xue, and W.-C. Siu, "Marker-based image segmentation relying on disjoint set union," *Signal Processing: Image Communication*, vol. 21, no. 2, pp. 100–112, 2006.

[73] P. R. Hill, C. N. Canagarajah, and D. R. Bull, “Image segmentation using a texture gradient based watershed transform,” *IEEE Trans. on Image Processing*, vol. 12, no. 12, pp. 1618–1633, 2003.

[74] H. T. Nguyen, M. Worring, and R. Van Den Boomgaard, “Watersnakes: Energy-driven watershed segmentation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 3, pp. 330–342, 2003.

[75] I. Vanhamel, I. Pratikakis, and H. Sahli, “Multiscale gradient watersheds of color images,” *IEEE Trans. on Image Processing*, vol. 12, no. 6, pp. 617–626, 2003.

[76] J.-B. Kim and H.-J. Kim, “Multiresolution-based watersheds for efficient image segmentation,” *Pattern Recognition Letters*, vol. 24, no. 1, pp. 473–488, 2003.

[77] H. Cheng and J. Li, “Fuzzy homogeneity and scale-space approach to color image segmentation,” *Pattern Recognition*, vol. 36, no. 7, pp. 1545–1562, 2003.

[78] S. B. Chaabane, M. Sayadi, F. Fnaiech, and E. Brassart, “Colour image segmentation using homogeneity method and data fusion techniques,” *EURASIP J. of Advances in Signal Processing*, vol. 2010, pp. 1–11, 2010.

[79] S.-K. Choy, M.-L. Tang, and C.-S. Tong, “Image segmentation using fuzzy region competition and spatial/frequency information,” *IEEE Trans. on Image Processing*, vol. 20, no. 6, pp. 1473–1484, 2011.

[80] A. Protiere and G. Sapiro, “Interactive image segmentation via adaptive weighted distances,” *IEEE Trans. on Image Processing*, vol. 16, no. 4, pp. 1046–1057, 2007.

[81] S. Xiang, F. Nie, C. Zhang, and C. Zhang, “Interactive natural image segmentation via spline regression,” *IEEE Trans. on Image Processing*, vol. 18, no. 7, pp. 1623–1632, 2009.

[82] X. Bai and G. Sapiro, “Geodesic matting: A framework for fast interactive image and video segmentation and matting,” *Int. J. of Computer Vision*, vol. 82, no. 2, pp. 113–132, 2009.

[83] H. Li and C. Shen, "Interactive color image segmentation with linear programming," *Machine Vision and Applications*, vol. 21, no. 4, pp. 403–412, 2010.

[84] L. Shi and B. Funt, "Quaternion color texture segmentation," *Computer Vision and Image Understanding*, vol. 107, no. 1, pp. 88–96, 2007.

[85] Ö. N. Subakan and B. C. Vemuri, "A quaternion framework for color image smoothing and segmentation," *Int. J. of Computer Vision*, vol. 91, no. 3, pp. 233–250, 2011.

[86] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Proc. 2006 European Conf. on Computer Vision (ECCV)*, (Graz, Austria), pp. 1–15, May 2006.

[87] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *Int. J. of Computer Vision*, vol. 81, no. 1, pp. 2–23, 2009.

[88] C. H. Wang and L. Guan, "Graph cut video object segmentation using histogram of oriented gradients," in *Proc. 2008 IEEE Int. Symposium on Circuits and Systems (ISCAS)*, (Seattle, WA, USA), pp. 2590–2593, May 2008.

[89] P. Nammalwar, O. Ghita, and P. F. Whelan, "A generic framework for colour texture segmentation," *Sensor Review*, vol. 30, no. 1, pp. 69–79, 2010.

[90] X. Li, H. Fan, Y. Zhao, and H. Zhang, "Graph cuts based image segmentation using local color and texture," in *Proc. 2011 Int. Congress on Image and Signal Processing (CISP)*, vol. 3, (Shanghai, China), pp. 1251–1255, Oct. 2011.

[91] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.

[92] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[93] X.-Y. Wang, T. Wang, and J. Bu, "Color image segmentation using pixel wise support vector machine classification," *Pattern Recognition*, vol. 44, no. 4, pp. 777–787, 2011.

[94] Z. Yu, H.-S. Wong, and G. Wen, "A modified support vector machine and its application to image segmentation," *Image and Vision Computing*, vol. 29, no. 1, pp. 29–40, 2011.

[95] L. Macaire, N. Vandenbroucke, and J.-G. Postaire, "Color image segmentation by analysis of subset connectedness and color homogeneity properties," *Computer Vision and Image Understanding*, vol. 102, no. 1, pp. 105–116, 2006.

[96] G. Delyon, F. Galland, and P. Réfrégier, "Minimal stochastic complexity image partitioning with unknown noise model," *IEEE Trans. on Image Processing*, vol. 15, no. 10, pp. 3207–3212, 2006.

[97] A. Bardera, J. Rigau, I. Boada, M. Feixas, and M. Sbert, "Image segmentation using information bottleneck method," *IEEE Trans. on Image Processing*, vol. 18, no. 7, pp. 1601–1612, 2009.

[98] Y.-Z. Song, B. Xiao, P. Hall, and L. Wang, "In search of perceptually salient groupings," *IEEE Trans. on Image Processing*, vol. 20, no. 4, pp. 935–947, 2011.

[99] D. Glasner, S. N. Vitaladevuni, and R. Basri, "Contour-based joint clustering of multiple segmentations," in *Proc. 2011 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, (Colorado Springs, CO, USA), pp. 2385–2392, Jun. 2011.

[100] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching-incorporating a global constraint into MRFs," in *Proc. 2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, (New York, NY, USA), pp. 993–1000, Jun. 2006.

[101] D. Weinland, R. Ronfard, and E. Boyer, “A survey of vision-based methods for action representation, segmentation and recognition,” *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[102] A.-I. Popa and C. Sminchisescu, “Parametric image segmentation of humans with structural shape priors,” *arXiv preprint arXiv:1501.06722*, 2015.

[103] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[104] T. Zhao and R. Nevatia, “Stochastic human segmentation from a static camera,” in *Proc. 2002 IEEE Workshop on Motion and Video Computing (MOTION)*, (Orlando, FL, USA), pp. 9–14, Dec. 2002.

[105] G. Mori, X. Ren, A. Efros, and J. Malik, “Recovering human body configurations: Combining segmentation and recognition,” in *Proc. 2004 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, (Washington, DC, USA), pp. 326–333, Jul. 2004.

[106] D. R. Martin, C. C. Fowlkes, and J. Malik, “Learning to detect natural image boundaries using local brightness, color, and texture cues,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004.

[107] V. Sharma and J. W. Davis, “Integrating appearance and motion cues for simultaneous detection and segmentation of pedestrians,” in *Proc. 2007 IEEE Int. Conf. on Computer Vision (ICCV)*, (Rio de Janeiro, Brazil), pp. 1–8, Oct. 2007.

[108] Z. Lin and L. S. Davis, “Shape-based human detection and segmentation via hierarchical part-template matching,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 604–618, 2010.

[109] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[110] S. Z. Li, "Markov random field models in computer vision," in *Proc. 1994 European Conf. on Computer Vision (ECCV)*, (Stockholm, Sweden), pp. 361–370, May 1994.

[111] S. Prince, *Computer Vision: Models Learning and Inference*. New York, NY, USA: Cambridge University Press, 2012.

[112] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive GMMRF model," in *Proc. 2004 European Conf. on Computer Vision (ECCV)*, (Prague, Czech Republic), pp. 428–441, May 2004.

[113] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models—Their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[114] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[115] C. Vachier and F. Meyer, "The viscous watershed transform," *J. of Mathematical Imaging and Vision*, vol. 22, no. 2-3, pp. 251–267, 2005.

[116] G. Bradski, "The OpenCV library," *Dr. Dobb's J. of Software Tools*, vol. 25, no. 11, pp. 120–126, 2000.

[117] T. Cour, S. Yu, and J. Shi, "Normalized cut segmentation code." `http://www.timotheecour.com/software/ncut/ncut.html`, 2004. Accessed 2013/01/05.

[118] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. 2005 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, (San Diego, CA, USA), pp. 886–893, Jun. 2005.

[119] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, (Kauai, HI, USA), pp. 511–518, Dec. 2001.

[120] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. 2011 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, (Colorado Springs, CO, USA), pp. 2097–2104, Dec. 2011.

[121] H. Zhou, J. Zheng, and L. Wei, "Texture aware image segmentation using graph cuts and active contours," *Pattern Recognition*, vol. 46, no. 6, pp. 1719–1733, 2013.

[122] N. Sochen, R. Kimmel, and R. Malladi, "A general framework for low level vision," *IEEE Trans. on Image Processing*, vol. 7, no. 3, pp. 310–318, 1998.

[123] L. Wang, J. Shi, G. Song, and I.-F. Shen, "Object detection combining recognition and segmentation," in *Proc. 2007 Asian Conf. on Computer Vision (ACCV)*, (Tokyo, Japan), pp. 189–199, Nov. 2007.

[124] P. Kontschieder, M. Fiterau, A. Criminisi, and S. Rota Bulo, "Deep neural decision forests," in *Proc. 2015 IEEE Int. Conf. on Computer Vision (ICCV)*, (Santiago, Chile), pp. 1467–1475, Dec. 2015.

Publications

Journals

[1] E. Pourjam, D. Deguchi,I. Ide and H. Murase, “Statistical shape feedback for human subject segmentation”, *IEEJ Trans. on Electronics, Information and Systems*, vol. 135, no. 8, pp. 1000–1008, 2015.

[2] E. Pourjam, D. Deguchi, I. Ide, and H. Murase, “Using super-pixels and human probability map for automatic human subject segmentation”, *IEICE Trans. on Fundamentals*, vol. E99-A, no. 5 (to be published in May 2016).

[3] S. Azadi, E. Pourjam, A. Nikkerdar, and M. Sharifi, “Real-time object tracking using gradient vector flow”, *Przegld Elektrotechniczny*, vol. 89, no. 1a, pp. 280–283, 2013.

Conferences (Reviewed)

[1] E. Pourjam, I. Ide, D. Deguchi, and H. Murase, “Segmentation of human instances using grab-cut and active shape model feedback”. in *Proc. 2013 IAPR Conf. on Machine Vision Applications (MVA)*, (Kyoto, Japan), pp. 77–80, May 2013.

[2] E. Pourjam, Y. Kawanishi, I. Ide, D. Deguchi, and H. Murase, “Human body segmentation with texture grab-cut and active shape models”, in *Proc. 2016 Frontiers of Computer Vision Workshop (FCVW)*, (Takayama, Japan), pp. 28–33, Feb. 2016.

Conferences (Non-reviewed)

[1] E. Pourjam, D. Deguchi, I. Ide, and H. Murase, “Automatic pedestrian segmentation using grab-cut and active shape model feedback”. in *Proc. 16th Meeting on Image Recognition and Understanding (MIRU)*, (Tokyo, Japan), no. SS2-12, Aug. 2013.