電気学会論文誌 C（電子・情報・システム部門誌）
IEEJ Transactions on Electronics, Information and Systems
Vol.135 No.8 pp.1000–1008 DOI: 10.1541/ieejeiss.135.1000

Paper

Statistical Shape Feedback for Human Subject Segmentation

Esmaeil Pourjam[*a)] Non-member, Daisuke Deguchi[**] Non-member
Ichiro Ide[*] Non-member, Hiroshi Murase[*] Non-member

(Manuscript received June 11, 2014, revised March 11, 2015)

Human segmentation is one of the most interesting yet most challenging subjects in the field of object segmentation and image processing. It can be used in various types of applications from image retrieval to robotics and human machine interfaces, including even entertainment. Many researches have been done on this subject and it is still one of active research areas. But until now, a method for accurate segmentation in different conditions has not been introduced. In this paper, we present "Statistical Shape Feedback Segmentation" (SSFSeg) method, which is a way to automatically segment human subjects (pedestrians) from single images. Our main contributions in this paper are: 1) Using human shape model as priors for Grab-cut segmentation. 2) Implementation of a feedback system which provides a coarse-to-fine way of generating more accurate shapes. For this task, we try to use masks generated by the Statistical Shape Model (SSM) algorithm as a prior input for the Grab-cut technique to segment the desired human subject in the image without user interaction. To achieve this, we propose a feedback framework for the SSM sample generation. Our experiments confirmed that the segmentation error of our proposed method is less than half of the Grab-cut method.

Keywords: Human Segmentation, Grab-cut, Statistical Shape Model

1. Introduction

Human segmentation is one of the most interesting yet most challenging subjects in the field of object segmentation and image processing. It can be used in various types of applications from image retrieval to robotics and human machine interfaces, including even entertainment. Many researches have been done on this subject and it is still one of active research areas. But until now, a method that can segment the subject-of-interest with high precision and is robust in different situations has not been introduced. There are a lot of problems that have to be solved, like environmental illumination changes, imaging noises and more importantly, the human body which is an articulated type of object, makes it very difficult to model. Especially since humans wear various types of clothing in different kinds of situations, the problem becomes more intense.

The object segmentation problem itself can be divided into two main categories: automatic [(1)–(7)] and interactive [(8)–(18)] 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 the user interactions in different levels of segmentation process to avoid miss-segmentations.

In recent years, interactive segmentation methods like "Lazy Snapping" [(8)], "TVSeg" [(10)], the work from Gulshan et al. [(11)], "Graph-cut" [(12)], "Grab-cut" [(18)], "Normalized-cut" [(19)] and "Watershed" [(20)] have become popular and also have shown some promising results, so if we become capable of utilizing these methods in an automatic manner, we can take advantage of their accuracy.

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 results, user input is necessary for achieving satisfactory results which renders them useless for automatic applications.

Here, one idea for creating an automatic segmentation system based on an interactive method is to use human detection algorithms like famous HOG detectors [(21)], cascade detectors [(22)] or detectors like the work of Benenson et al. [(23)] which gives us the place of a human subject in the image. We can use the result of the detection to select a rectangle around a subject and his/her surrounding area and use this selection as an input for interactive segmentation methods like Grab-cut which just needs a rectangle around the object initialization. Though this idea is feasible, the main problem is that, even with Grab-cut that has relatively accurate results when used interactively, just giving the rectangle around a human subject and using automatic segmentation will not result in a good segmentation unless the human subject and his/her surrounding have distinctly different color distributions (the background of the subject is relatively simple or is blurred), which is not correct in most of the cases.

To overcome these problems, in this paper we propose the following points:

a) Correspondence to: Esmaeil Pourjam. E-mail: esmaeilp@murase.m.is.nagoya-u.ac.jp
[*] Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8601, Japan
[**] Information Strategy Office, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8601, Japan

(1) Using 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.

By using human body shape model, we gain the ability to cope with various body deformations, resulting in more robustness and accuracy compared to the conventional methods. As one of the ways for modeling human body, here we use the statistical shape model (SSM) algorithm and use shapes generated by that as prior information for our modified Grab-cut based segmentation method.

We also propose a feedback system based on SSM shape generation, introducing a coarse-to-fine shape generation procedure which refines the generated shapes step by step, making 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 the paper is organized as follows: First we give a brief review of some related works in Section 2. Section 3 will explain the details of our proposed method. Some experiments have been performed for testing the validity of the proposed method which will be presented in Section 4. There would be a discussion about system implementation and parameters selection in Section 5, and finally we will conclude our work with Section 6.

2. Related Works

Many segmentation methods exist in the literature each with their own benefits and shortcomings. There also exist some surveys like that by Weinland et al.[24] that explain famous methods, so here we just introduce some of the methods which are somehow related to our work.

2.1 Automatic Segmentation Recently, Zhang et al.[4] have proposed a video object segmentation method which uses Directed Acyclic Graph (DAG) and graph-based algorithm for unsupervised segmentation of primary object in a video sequence. They first try to extract some proposals for the main object in the current frame of video using DAG and also use optical flow and selected proposal for the previous frame to predict the main object in the current frame. Then they use the extracted proposals and the prediction for expanding the proposal set. Using these proposals, they try to segment the primary object in the video. The main problem of their work is it uses the information in multiple frames of the video so it is not possible to use it with just one image. By realizing single frame segmentation, the system would be applicable for a wider range of applications.

Gulshan et al.[6] have taken the advantage of Microsoft Kinect and tried to propose an automatic segmentation algorithm. They first create a training dataset based on images acquired from Kinect (depth map & image together). After that they extract HOG features from images in the data set and use them for training a classifier. When a new image is input to the system, this classifier generates a rough segmentation which is then given to a local Grab-cut stage for more precise segmentation. The main problem is that the system is in need of a large set of images for training the classifier (1,930 images used in their case). Also the segmentation has become a two-step process where Grab-cut is used for local refinement, so in case either of the steps fails, the whole result would be affected.

Prakash et al.[13] apply active contour (snakes) algorithm, one of traditional methods of segmentation in conjunction with Grab-cut to increase the segmentation accuracy. Their system uses Grab-cut to segment inside parts while the active contour segments the boundaries of the object. Since their system uses the active contour and Grab-cut in parallel and combines their segmentation results as output, if either of the steps fail, the output would be affected, e.g. if active contour fails to find correct object boundaries or if Grab-cut fails due to color similarity of object with background, the result might not be satisfactory.

2.2 Interactive Segmentation Kuang et al.[14] try to learn two image features (color and texture) and a smoothing parameter from two polygons drawn by the user as seeds for foreground and background. Their method maximizes a weighted energy function margin for estimating the parameters iteratively and at the same time segments the image. The interesting point in their research is that the system will learn optimized parameters specific to each input image. But since the user must specify the initializing seeds for foreground and background, if this selection is not good enough or the object has different color distributions which are not included in the seeds, the segmentation result might not be satisfactory.

Li et al.[15] present a framework for segmenting objects in video sequences. In their work, a 3D graph-cut based segmentation is proposed based on the precise segmentation in the key frames. They also provide the user a way to correct the miss-segmentations in local frames. Since the system needs the precise segmentation of the object-of-interest in key frames by the user, depending on the number of key frames (usually sampled each ten frames as they mentioned in their work) a lot of work might be needed aside from the corrections for miss-segmentations by the system.

Peng & Veksler[16] use a training set with different segmentation results of images (ten segmentations per image) manually labeled as "good" or "bad" to train an AdaBoost based classifier. After the user inputs all background and foreground seeds, the system tries to find a result, classified as a most confidently "good" segmentation. The user then may input some corrections and rerun the program to achieve better results. Thus the final result of the system is highly dependent on the accuracy of the training data (how accurately the images labeled with "good" are good segmentation), the classifier performance and user corrections.

Szummer et al.[17] try to learn segmentation parameters automatically using structured support vector machine (SVM_{STRUCT}) and maximum-margin network learning. In their work, the user selects a polygon depicting the rough region of a foreground object and the system iteratively learns the parameters and segments the image. As their parameter learning system at each iteration just adds one solution to a solution set, the rate of convergence of the system can become slow depending on the situation.

Rother et al.[18] introduce Grab-cut segmentation which is an upgraded model of the famous graph-cut segmentation[12], incorporating the color features and a better iterative energy

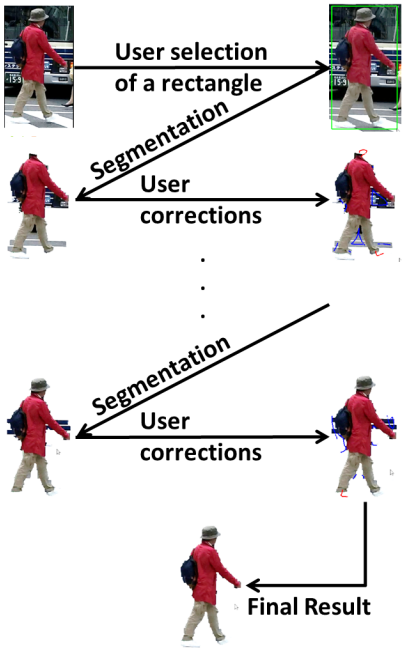


Fig. 1. Basic process flow in the Grab-cut segmentation framework.

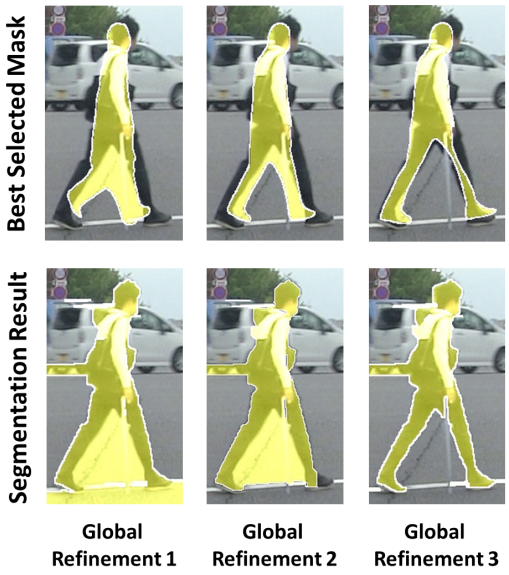


Fig. 2. An example of step-by-step mask refinement. The top row shows the best locally selected mask at each stage. The bottom row shows the segmentation result using that mask.

minimization procedure. As it can be seen in Fig. 1, the user just needs to select a rectangle around the object-of-interest. However, there is one main problem here, this method cannot segment the image completely just by itself and relies on the user for further foreground and background seeds selection.

Aside from the problems mentioned above, we can say that the biggest problem with the methods presented in this part is that all of them are in need of manual user initialization and/or correction for achieving the final segmentation result.

3. Statistical Shape Feedback Segmentation

3.1 Main Idea Two ideas proposed in this paper 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 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 our work 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 deformability 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 human body shape is possible because even if the body is highly deformable, since there are some physical constraints on the body, the degree of the shape variations is limited, so it is possible to model the shape changes in a mathematical way. So 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.

In this work, we try to exploit the mentioned idea for our segmentation system. As one of the ways to model the shape, we chose the SSM method to model human body shape and use the generated samples from a system, trained with real pedestrian samples, as a basis for human subject segmentation. Our aim is to use the flexibility of SSM algorithm for generating new shapes in addition to segmentation accuracy of Grab-cut and propose a system which can segment human subjects automatically and accurately.

Although we can 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 we want to segment, and as a result, just by generating shapes we might not get 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, also it 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 one time, 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 Fig. 2.

We call our proposed method “Statistical Shape Feedback Segmentation” and in the rest of the paper, we will use the abbreviated form “SSFSeg” to refer to it. The general process flow of the system is shown in Fig. 3.

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

- **SSM Generation Step**
  - Some new samples based on the training data are generated.

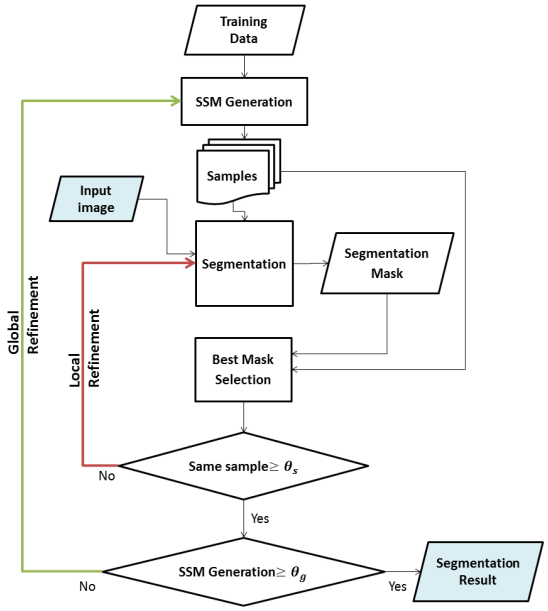


Fig. 3. Process flow of the proposed SSFSeg method.

- **Mask Generation Step**
  - The selected sample is converted into a trimap.
- **Segmentation Step**
  - (1) Image containing the 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 segmentation result is stabilized, finish the procedure and show the result, else start over from the SSM generation step.

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

3.2 SSM Generation Although there are many ways to model a human body, like active shape models (ASM) or active appearance models (AAM), here we use statistical shape model (SSM) as the method for encoding our training samples into a mathematical model and use it for our segmentation system.

In the first step, we start by generating some new shapes based on the training dataset using the conventional SSM method, first introduced by Coots et al. [25]. 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, we segment some training images 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 object with varying shapes can also be used). 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 we can align them as described by Cootes et al. [25]. Thus for each image, we will have a vector with $2n$ points like:

$$\mathbf{x}_i = [x_1, y_1, ..., x_n, y_n]^T \cdots\cdots (1)$$

Now, we can calculate the mean model for the shape domain as:

$$\overline{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i \cdots\cdots (2)$$

Based on these, we can calculate the covariance matrix:

$$\boldsymbol{S} = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T \cdots\cdots (3)$$

By analyzing this $2n \times 2n$ matrix and calculating its eigenvalues (λ_i) and 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:

$$x_{\text{new}} = \overline{\mathbf{x}} + \boldsymbol{P}\mathbf{b} \cdots\cdots (4)$$

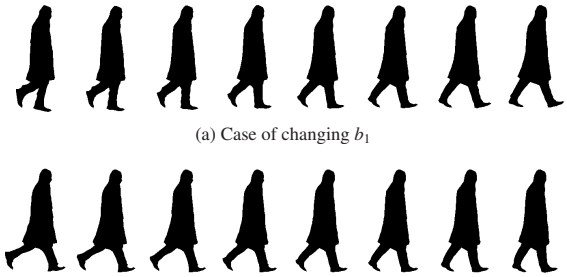
Here, matrix $\boldsymbol{P}$ is made by setting the selected eigenvectors as columns, and $\mathbf{b}$ is a vector of weights like:

$$\boldsymbol{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_t] \cdots\cdots (5)$$
$$\mathbf{b} = [b_1, \ldots, b_t] \cdots\cdots (6)$$

A suitable limit for the weights can be described as:

$$-2.5\sqrt{\lambda_k} \le b_k \le 2.5\sqrt{\lambda_k},\ k \in [1, \ldots, t] \cdots\cdots (7)$$

In Fig. 4, 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, $\mathbf{b} = [b_1, 0, \ldots, 0]^T$.



(a) Case of changing b_1

(b) Case of changing b_2

Fig. 4. Some samples generated with SSM.

3.3 Mask Generation After new shapes from the SSM generation step are obtained, we have to somehow use them as prior information for segmentation.

For this, we make a trimap of labels for initialization of the segmentation step and also for further segmentations in local and global refinement stages. For labeling purposes, we use three types of labels:

- 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 being part of foreground is more than being part of 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: Like the "Probably foreground" but defined for background pixels (this time the probability of this part being part of background is higher).

To make the trimap, we first use erosion binary operator to create the main part of the mask, this part is labeled as "Foreground" in the trimap. Since our human subject would not have the same shape as our generated mask, we should somehow tell the system to search the image in an area more than the one indicated by the mask itself. We do this 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 trimap. Figure 5 shows an example of a generated mask and the trimap created from it.

3.4 Segmentation Now that we have a generated shape and have already converted it to a trimap, we can start our segmentation procedure. In this work, we will use our previous work [26] as a basis for the segmentation system by modifying the Grab-cut method introduced by Rother et al. [18] so we can use our generated shapes as priors for segmentation.

Like most other segmentation algorithms, in Grab-cut, the object segmentation is defined as an energy minimization problem in the following form:

$$\mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) + V(\underline{\alpha}, \mathbf{z}) \cdots\cdots\cdots\cdots (8)$$

which consists of an unary term

$$U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = \sum_n D(\alpha_n, k_n, \underline{\theta}, z_n) \cdots\cdots\cdots\cdots\cdots (9)$$

where

$$D\left(\alpha_n, k_n, \underline{\theta}, z_n\right) = -\log p(z_n \mid \alpha_n, k_n, \underline{\theta}) - \log \pi(\alpha_n, k_n) \cdots\cdots\cdots (10)$$

And a smoothness term

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n)\in \mathbf{C}} \mathrm{dist}^{-1}(m,n)[\alpha_n \neq \alpha_m] \exp^{-\beta \|z_m - z_n\|^2} \cdots (11)$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ is the RGB color values of the image, $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)$ is an array of opacity values

(a) Generated mask (b) Resulted trimap

Fig. 5. Converting a generated sample to a trimap of "Foreground" (light gray), "Probably foreground" (gray), and "Probably background" (dark gray).

Table 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

$0 \leq \alpha_n \leq 1$, but for hard segmentation it is either 1 (foreground) or 0 (background). $\underline{\theta}$ shows foreground and background models expressed by GMMs.

$$\underline{\theta} = \{W(\alpha,k), \Gamma(\alpha,k), \Sigma(\alpha,k), \alpha = 0,1, k = 1 \ldots K\} \cdots\cdots\cdots\cdots\cdots (12)$$

W, Γ and Σ are weights, means and covariences respectively. $p(\cdot)$ is Gaussian probability distribution, and $\pi(\cdot)$ are mixture weighting coefficients, β is defined by Boykov and Jolly [12] like

$$\beta = \frac{1}{2\langle (z_m - z_n)^2 \rangle} \cdots\cdots\cdots\cdots\cdots\cdots\cdots (13)$$

Setting β to zero results in under-segmentations while setting it to big values will lead to some over-segmentations, and as a result, γ is experimentally selected and set to 50 in (18).

Since our energy model is now complete, it is possible to use the standard minimum-cut algorithm to estimate a hard segmentation iteratively.

Although the Grab-cut algorithm used here is almost the same as in the original paper [18], we slightly modified the smoothness parameter by adding a distance penalty between the generated mask boundary and input image pixels thus the parameter γ in the original Grab-cut paper [18] which is a constant value becomes variable relative to the minimum distance of each image pixel to the boundary of the generated mask from the SSM stage. So we can rewrite the smoothness term of Eq. (11) in form of

$$V(\underline{\alpha}, \mathbf{z}) = \gamma(m) \sum_{(m,n)\in \mathbf{C}} \mathrm{dist}^{-1}(m,n)[\alpha_n \neq \alpha_m] \exp^{-\beta \|z_m - z_n\|^2} \cdots\cdots\cdots\cdots\cdots (14)$$

in which

$$\gamma(m) = \gamma_0 + \gamma_1 \times \mathrm{dist}(m, m') \cdots\cdots\cdots\cdots\cdots (15)$$

where $[\cdot]$ is indicator function and m' is the nearest point on the boundary of the generated mask to point m in the image and $\mathrm{dist}(m,n)$ is Euclidean distance function. The result of the distance function is also normalized so that difference between point distances does not become dominant in the smoothness term. The value above is also selected based on the experiments with different sets of images. Table 1 shows how assigning different values to the parameter affects the final segmentation accuracy.

3.5 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

Fig. 6. Some samples from the testing dataset containing 180 pedestrian images.

Table 2. Size and processing time for some of the sample images in Fig. 6

Image No.	1	2	3	4	5	6	7	8	9	10	11	12	13
Size (pixels)	94×216	150×217	150×241	93×234	207×393	198×320	83×127	72×104	56×105	116×221	59×100	150×242	246×412
Time (sec.)	16.83	33.32	41.29	13.30	151.55	82.06	6.36	5.55	10.15	16.90	4.26	31.70	169.59

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).

3.6 Global Refinement After 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 2 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, we here use the generated sample as a ground truth for segmentation provided that the desired object should be similar to the provided mask to some degree.

4. Experiments

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

4.1 Dataset Two datasets have been used for our experiments. There are already different pedestrian datasets available which are made from videos taken by in-vehicle cameras. These datasets include different human subjects with various poses and different situations. As a result, for this work we tried to evaluate our method with both a private dataset and the famous Caltech pedestrian dataset which are created based on mentioned type of videos.

The first dataset used for testing the system in this paper is a private set of 180 images from different human subjects (full body) in different situations which we created based on data available in our laboratory. All images are taken with an in-vehicle camera and are color images with different sizes from 47×80 pixels to 378×618 pixels. The images are all taken in the day time and night time images are not included in either training nor testing experiments. Some samples of the test dataset are presented in Fig. 6, with their size and processing time in Table 2.

The second dataset is collection of 100 images taken from Caltech pedestrian dataset [27]. Images have different sizes and are all taken in day time. Images of pedestrians with a height more than 60 pixels have been selected randomly for testing the proposed system.

As for the SSM dataset, 60 samples for training the SSM model have been used which are not included in either of the test datasets. All samples are hand-segmented silhouettes of real pedestrians selected from images taken by in-vehicle camera like the test dataset. Training image size is 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 testing datasets.

4.2 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 stop the segmentation and the generation process, experiments show that if we set $\theta_s = 5$ and $\theta_g = 3$, as it can be viewed from Table 1, usually desired results would be achieved.

4.3 γ Parameter Selection It can be seen in Table 1 that if we set $\gamma_0 = 0.07$ and $\gamma_1 = 75$, we would obtain the best results so in all of the tests for the proposed method, these values have been selected for $\gamma(m)$. Thus we have set

$$\gamma(m) = 0.07 + 75 \times \mathrm{dist}(m, m') \cdots\cdots\cdots\cdots\cdots\cdots (16)$$

4.4 Results The comparison is done between the original Grab-cut segmentation [18], Normalized Cut (N-cut) segmentation [19], Watershed [20] segmentation and the pro-

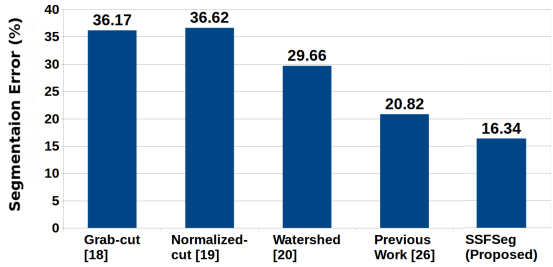


Fig. 7. Comparison between the proposed method and other segmentation methods using private dataset.

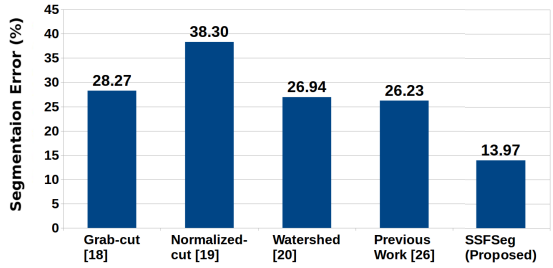


Fig. 8. Comparison between the proposed method and other segmentation methods using images from Caltech pedestrian dataset (27).

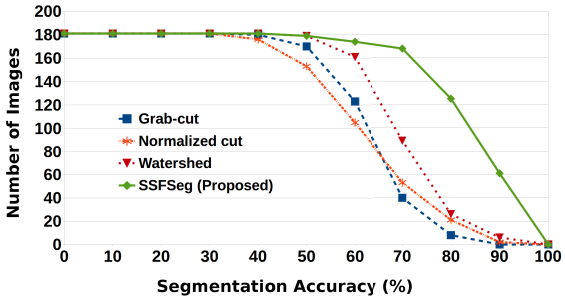


Fig. 9. Performance of the proposed and the comparative methods. Number of dataset images that were segmented with accuracy over 70% by the proposed SSFSeg method is much higher compared to comparative methods.

posed 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:

$$\text{Error } (\%) = \frac{\text{FN} + \text{FP}}{\text{Number of pixels in the image}} \times 100 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (17)$$

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 (28) open source library, and for Normalized-cut segmentation, the code provided by (29) were used.

Figures 7 and 8 show segmentation error from using the proposed method and other comparative methods. As it can be seen in the image, the segmentation error is significantly decreased compared to other methods (cut by half comparing

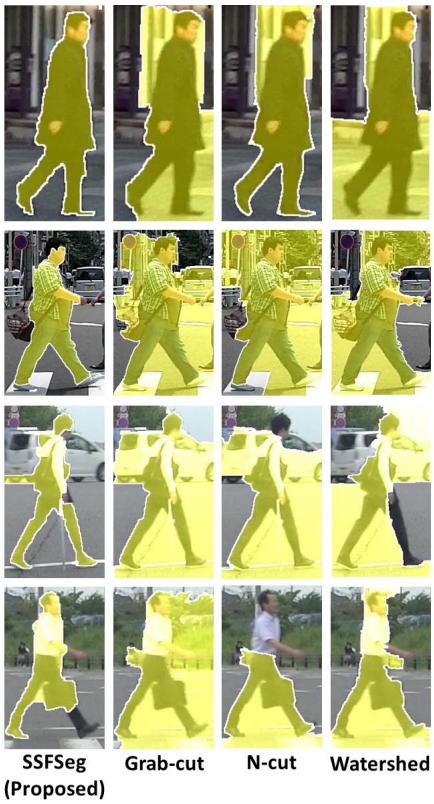


Fig. 10. Example of image segmentation results. Segmentation results are overlaid on the input image with yellow color.

to the Grab-cut and Normalized-cut).

There is also Fig. 9 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 our proposed method, 168 images out of 181 images in the dataset are segmented with accuracy more than 70% while this number is 40 for Grab-cut, 53 for Normalizd-cut and 89 for Watershed.

Figure 10 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.

5. Discussion

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

As for θ_g and θ_s, Fig. 11 shows the results of repeating the SSM generation step from one time to 10 times and local refinement from one to 10 times. As it can be seen, the best result is achieved when we set $\theta_s = 5$ and $\theta_g = 3$.

Although in overall process, changing these values affect the final segmentation result within a 3% range, but 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 1. As the table shows, in the experiments when we set the γ parameters like Eq. (16), the segmentation error is decreased to its minimum value (16.34% in the segmenta-

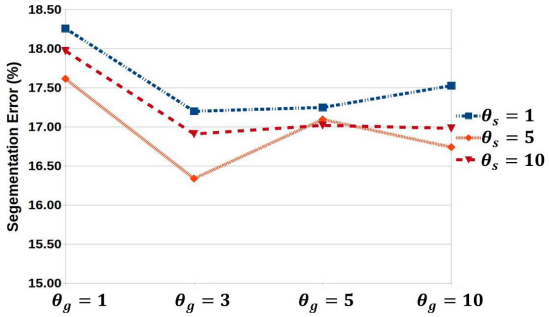


Fig. 11. Relation between θ_s and θ_g, and the effect of changing their values on segmentation error.

Fig. 12. Result of using feedback. The first row shows the segmentation results without using feedback while the second row shows the results using feedback.

tion 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 as similar as possible to the segmentation result thus improving the final segmentation result. Effect of using feedback in the system can be seen in Fig. 12.

The system has been implemented in C++ and is not optimized at current stage. Also it uses single thread for computation purposes. Table 2 shows the computation time required for processing the images in Fig. 6. We think that by code optimization, it is possible to reduce the time consumption significantly.

Still there are also some miss-segmentation cases as shown in Fig. 13. The problem in the first row is mainly because of similarity between foreground and background colors. Meanwhile, the second row shows the miss-segmentation because of wrong seed selection which we intend to solve in our future works.

It is also good to note that the proposed system uses and generates full body silhouettes at SSM stage so it does not consider the case of occlusions which is one of the cases we want to include in our future works. The method explained in this paper expects the output of a human detector algorithm (e.x. (21) and (22)) 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.

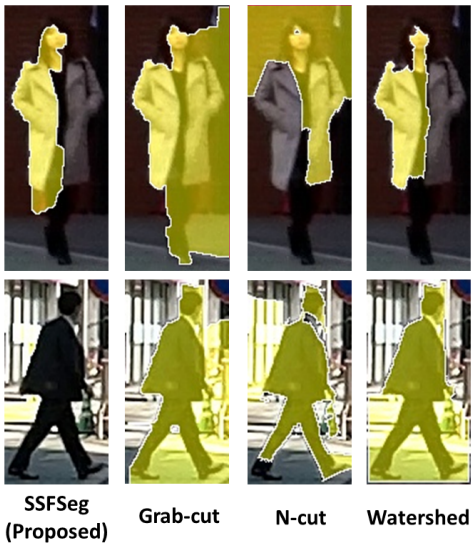


Fig. 13. An example of miss-segmentation in the proposed method and equivalent segmentation in comparative methods.

6. Conclusion

In this paper, we presented a method that can perform segmentation of pedestrians automatically and with more accuracy. 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 segmentation step instead of asking the user to identify the background and foreground seeds.

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. For example, 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.

As for the future work, we would like to:

- Make a more complete training dataset for the SSM generation step which includes more variations in the model.
- Use more features (color, texture, etc.) for modeling the object features.
- Optimize the code, since the time consumption is one of the problems here.
- Extend the algorithm and devise a multi-frame segmentation scheme.

Acknowledgment

Parts of this research were supported by MEXT, Grant-in-Aid for Scientific Research. Also parts of work were developed based on the MIST library from Nagoya University (http://mist.murase.m.is.nagoya-u.ac.jp/).

References

(1) M. Kass, A. Witkin, and D. Terzopoulos: “Snakes: Active contour models”, *International Journal of Computer Vision*, Vol.1, No.4, pp.321–331 (1988)

(2) V. Caselles, R. Kimmel, and G. Sapiro: “Geodesic active contours”, *International Journal of Computer Vision*, Vol.22, No.1, pp.61–79 (1997)

(3) T.F. Chan and L.A. Vese: “Active contours without edges”, *IEEE Trans. on Image Processing*, Vol.10, No.2, pp.266–277 (2001)
(4) D. Zhang, O. Javed, and M. Shah: “Video object segmentation through spatially accurate and temporally dense extraction of primary object regions”, In Proc. 26th IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, pp.628–635 (2013)
(5) 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)
(6) V. Gulshan, V. Lempitsky, and A. Zisserman: “Humanising Grabcut: Learning to segment humans using the Kinect”, In Proc. 13th IEEE Int. Conf. on Computer Vision Workshops, pp.1127–1133 (2011)
(7) M.P. Kumar, P.H.S. Torr, and A. Zisserman: “Obj Cut”, In Proc. 18th IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Vol.1, pp.18–25 (2005)
(8) 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)
(9) E.N. Mortensen and W.A. Barrett: “Intelligent scissors for image composition”, In Proc. 22nd Conf. on Computer Graphics and Interactive Techniques, pp.191–198 (1995)
(10) M. Unger, T. Pock, W. Torbin, D. Cremers, and H. Bischof: “TVSeg—Interactive total variation based image segmentation”, In Proc. 19th British Machine Vision Conf., Vol.2, pp.335–354 (2008)
(11) V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman: “Geodesic star convexity for interactive image segmentation”, In Proc. 23rd IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, pp.3129–3136 (2010)
(12) Y.Y. Boykov and M.-P. Jolly: “Interactive graph cuts for optimal boundary & region segmentation of objects in ND images”, In Proc. 8th IEEE Int. Conf. on Computer Vision, Vol.1, pp.105–112 (2001)
(13) S. Prakash, R. Abhilash, and S. Das: “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)
(14) Z. Kuang, D. Schnieders, H. Zhou, K.-Y.K. Wong, Y. Yizhou, and B. Peng: “Learning image-specific parameters for interactive segmentation”, In Proc. 25th IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, pp.590–597 (2012)
(15) 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)
(16) B. Peng and O. Veksler: “Parameter selection for graph cut based image segmentation”, In Proc. 19th British Machine Vision Conf., pp.160–170 (2008)
(17) M. Szummer, P. Kohli, and D. Hoiem: “Learning CRFs using graph cuts”, In Proc. 10th European Conf. on Computer Vision, Part 2, Lecture Notes in Computer Science, Vol.5303, pp.582–595 (2008)
(18) 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)
(19) 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)
(20) C. Vachier and F. Meyer: “The viscous watershed transform”, *Journal of Mathematical Imaging and Vision*, Vol.22, No.2–3, pp.251–267 (2005)
(21) N. Dalal and B. Triggs: “Histograms of oriented gradients for human detection”, In Proc. 18th IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Vol.1, pp.886–893 (2005)
(22) P. Viola and M. Jones: “Rapid object detection using a boosted cascade of simple features”, In Proc. 14th IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Vol.1, pp. 511–518 (2001)
(23) R. Benenson, M. Mathias, R. Timofte, and L. Van Gool: “Pedestrian detection at 100 frames per second”, In Proc. 25th IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, pp. 2903–2910 (2012)
(24) 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)
(25) 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)
(26) E. Pourjam, I. Ide, D. Deguchi, and H. Murase: “Segmentation of human instances using Grab-cut and active shape model feedback”, In Proc. 13th IAPR Int. Conf. on Machine Vision Applications, pp.77–80 (2013)
(27) 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)
(28) “Open source computer vision library”, http://www.opencv.org/, retrieved March 2014.
(29) T. Cour, S. Yu, and J. Shi: “Normalized cut segmentation code”, http://www.timotheecour.com/software/ncut/ncut.html, retrieved March 2014.

Esmaeil Pourjam (Non-member) received his BEng in Electronics form Shahid Beheshti University of Tehran, Iran in 2009 and MEng in Mechatronics from Semnan University, Iran in 2011 and is currently pursuing his PhD in Information Science as a MEXT scholarship student in Graduate School of Information Science of Nagoya University, Japan. His main interests are robotics, computer vision and intelligent systems. His research is currently focused on human and pedestrian segmentation methods with applications in intelligent vehicles and human-machine interfaces.

Daisuke Deguchi (Non-member) received his BEng and MEng degrees in Engineering and a PhD degree in Information Science from Nagoya University, Japan, in 2001, 2003, and 2006, respectively. He is currently an Associate Professor in Information Strategy Office, Information & Communications, Nagoya University, Japan. He is working on the object detection, segmentation, recognition from videos, and their applications to ITS technologies, such as detection and recognition of traffic signs.

Ichiro Ide (Non-member) received his BEng, MEng, and PhD from The University of Tokyo in 1994, 1996, and 2000, respectively. He became an Assistant Professor at the National Institute of Informatics, Japan in 2000. Since 2004, he has been an Associate Professor at Nagoya University. He had also been a Visiting Associate Professor at National Institute of Informatics from 2004 to 2010, an Invited Professor at Institut de Recherche en Informartique et Systèmes Aléatoires (IRISA), France in 2005, 2006, and 2007, a Senior Visiting Researcher at ISLA, Instituut voor Informatica, Universiteit van Amsterdam from 2010 to 2011. His research interest ranges from the analysis and indexing to retargeting of multimedia contents, especially in large-scale broadcast video archives, mostly on news, cooking, and sports contents. He has been serving on program committees at conferences such as ACMMM, ICME, CVIM, MMM, and ICMR. He is a senior member of IEICE and IPS Japan, and members of JSAI, ANLP, IEEE, and ACM.

Hiroshi Murase (Non-member) received the BEng, MEng, and PhD degrees in Electrical Engineering from Nagoya University, Japan. In 1980 he joined the Nippon Telegraph and Telephone Corporation (NTT). From 1992 to 1993 he was a visiting research scientist at Columbia University, New York. From 2003 he is a professor of Nagoya University, Japan. He was awarded the IEICE Shinohara Award in 1986, the Telecom System Award in 1992, the IEEE CVPR (Conference on Computer Vision and Pattern Recognition) Best Paper Award in 1994, the IPS Japan Yamashita Award in 1995, the IEEE ICRA (International Conference on Robotics and Automation) Best Video Award in 1996, the Takayanagi Memorial Award in 2001, the IEICE Achievement Award in 2002, and the Ministry Award from the Ministry of Education, Culture, Sports, Science and Technology in 2003. Dr. Murase is IEEE Fellow, IEICE Fellow, and a member of the IPS Japan.