Human Body Segmentation Using Texture Aware Grab-cut and Statistical Shape Models

Esmaeil Pourjam*, Yasutomo Kawanishi*, Ichiro Ide*, Daisuke Deguchi†, Hiroshi Murase*
*Graduate School of Information Science, Nagoya University
Email: esmaeilp@murase.m.is.nagoya-u.ac.jp, {kawanishi,ide,murase}@is.nagoya-u.ac.jp
†Information Strategy Office, Nagoya University
Email: ddeguchi@nagoya-u.jp

***Abstract*—Segmentation is one of active areas in computer vision field with application in many areas from entertainment to intelligent vehicles (IVs). Among the objects, humans themselves have always been among the most interested subjects because of their special features.**

Since human body has an articulated structure, modeling and recognizing different variations in the body has proved to be very difficult. Wearing various kinds of clothes in different situations which can have a completely dissimilar appearance based on the clothing type, makes the modeling much more difficult. Add to this, the common problems of vision like illumination changes, blurring due to camera movements, etc. make the problem even more difficult. Thus having a system that can segment human subjects accurately can be useful in many applications.

In this paper, we propose a system for segmenting human subjects using Statistical Shapes Models (SSM) feedback and a texture aware version of Grab-cut which incorporates texture feature for improving the segmentation accuracy. Our experiments show that the proposed system has an acceptable accuracy compared to the state-of-the art interactive methods and much better than the conventional ones.

I. Introduction

For a long time, human subject segmentation has been one of active areas in the field of vision. Making systems and machines capable of interacting with humans or understanding the presence of human can be very useful in various types of applications. Face recognition for security systems, pose estimation, intention recognition for Human Machine Interfaces or Intelligent Vehicles (IVs) are some samples among the list of imaginable applications for human segmentation. Especially, recent trends in IVs have focused on driver assistance systems, object avoidance systems, and automatic driving systems in which understanding pedestrians presence and their intention can have a great impact on their performance.

Object segmentation methods can be categorized into two major categories: automatic, and interactive (semi-automatic) algorithms. The difference is that in automatic methods, usually human interference is unnecessary while it is opposite in interactive ones. Still, in contrast to automatic segmentation, in recent years, interactive image segmentation has shown some potential in the field of segmentation. Different methods have already been introduced in the literature such as graph-cut [1], obj-cut [2], lazy snapping [3], intelligent scissors [4], Grab-cut [5], TVSeg [6] and Geodesic matting [7]. Among these, those utilizing the Markov random field framework (graph-cut, Grab-cut, etc.) have shown more potential in comparison with the others. Based on that, some works like [8], [9], [10] have studied the application of this framework for automatic segmentation.

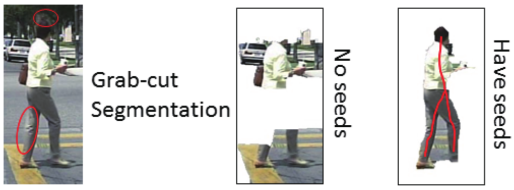
One of major problems when targeting human body is the large variations in its shape. This makes the task of modeling difficult and has led to different types of models to be introduced in the literature from simple skeletal to sophisticated probabilistic shape or part models. In addition, the variations of human clothings make the task much more complex because aside from the numerous combination to the texture and color of the cloths changes in the illumination can affect the perceived color to a great degree. As a result, many of the methods that use color as their primary parameter for differentiating between foreground and background like Graph-cut [1], Grab-cut [5], Gulshan et al.'s method [8], and some others will have serious problems in such cases. The problem also manifests itself when the texture of different regions on the body changes along side the color as depicted in Fig. 1.

In this paper, we propose an automatic system for segmenting human subjects using Statistical Shapes Models (SSM) feedback and an upgraded version of Grab-cut which incorporates texture feature. By incorporating color, shape, and texture factors, we show that the segmentation accuracy of the system can be improved and the above mentioned problems can also be avoided. Our experiments shows that the proposed system has an acceptable accuracy compared to the state-of-the art interactive methods and much better than the conventional ones.

The paper is organized as follows: In section 2, we will consider some background researches done in this field. Section 3 will explain the proposed method alongside explanations on statistical shape models, texture feature and modified Grab-cut. Section 4 will show the experimental results of the proposed method.

II. Related Works

As mentioned earlier, various segmentation methods exist in the literature. In this section, first some related methods will be reviewed. After that, a brief introduction to SSM and Grab-cut will be presented.



(a) (b) (c)

Fig. 1. An example of how using just color factor for segmentation may fail in Grab-cut [5], (a) input image in which some parts have similar color distribution to the background. (b) segmentation result using just a rectangle around the subject of interest, (c) adding some foreground seeds to the selected region.

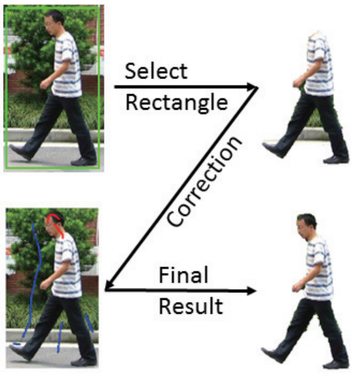


Fig. 2. An example of the segmentation process by the Grab-cut method.

A. Interactive Segmentation

Rother et al. [5] in 2004, introduced the now famous Grab-cut segmentation algorithm which tries to segment the foreground object based on a simple polygon (usually a rectangle) drawn by the user. For this, their system first learns two Gaussian Mixture Models (GMMs); one for foreground and one for background based on user selection. By turning the image into a graph and using max-flow/min-cut method iteratively, these models are refined until the object is segmented. The user can make some corrections in case of need. An example of the segmentation process by the Grab-cut method is presented in Fig. 2. M. Tang et al. [11] have tried to replace the energy model being used in Graph-cut based methods like Grab-cut [5] with a new, much simpler energy formula and also proposed an L_1 distance measure for minimizing the appearance overlap between a foreground object and its background. Utilizing this, their method segments a foreground object in one iteration and iterative energy minimization like the one proposed in Grab-cut is unnecessary.

Kuang et al. [12] use user input polygons (one for foreground, one for background) to learn color feature, texture feature and a smoothing parameter from the input image. Their method maximizes a weighted energy function margin for estimating the parameters iteratively, and at the same time segments the image. The notable point of this method is that, it learns aforementioned parameters (color, texture, smoothness) specific to the input image. This is mentioned in the work to be much better than pre-setting parameters by training the system beforehand.

(a) (b)

Fig. 3. Difference between using just color and using color & texture information. (a) Grab-cut (just color). (b) Proposed method (color & texture).

B. Automatic Segmentation

Zhang et al. [13] have proposed a video object segmentation method using a Directed Acyclic Graph (DAG) for object detection and segmentation. They assume as a basic principal that generally objects can be described with locally smooth motion trajectories, and try to find the primary object in the image series from available proposals. The proposals are created using optical flow, and by using dynamic programming, they try to find the best candidate in series of images.

Gulshan et al. [8] have utilized Microsoft Kinect for generating a training dataset, using depth information. They then used the generated dataset to train a classifier and proposed an automatic segmentation algorithm. For that they first extract HOG features from images in the dataset and then train a classifier. In the segmentation stage, the classifier generates a rough segmentation of the input image which is then refined using a local Grab-cut segmentation.

Prakash et al. [14] proposed an automatic object segmentation system by combining Active contour (snakes) [15] with the Grab-cut algorithm. The segmentation process is divided into two parallel procedures in their work; The active contour will try to find the boundary of the object from outside while the grab-cut tries to do it from inside. By combining the results of the two procedures, they then show that the segmentation accuracy increases in comparison with both original methods.

III. Texture Aware Grab-cut Segmentation

A. Main Idea

Trying to segment human subjects (other objects are the same) just based on the color distribution (like the color distance used by grab-cut) would not always result in desired results. Especially, since the human body is composed of different parts and humans wear various clothes, using just color for segmentation will lead to miss-segmentations like the one presented in Fig. 3. On the other hand, even if the color changes under different illuminations, the texture is known to be invariant. Also as mentioned in the work of Zhou et al. [16], texture is of a semi-local nature, so using it in combination with color information could be a very helpful asset in segmentation.

As a result, in this work, we try to propose a segmentation method that uses a modified version of Grab-cut method by incorporating texture feature in the framework and further improve the results using our previous method [17], [18]. The main idea here is to use the texture feature proposed by Zhou et al. [16] and create a texture map of the input image and use it as an auxiliary channel of information for the Grab-cut system. Also by using the method proposed in [18] for making a shape model generation and refinement, the proposed method becomes an automatic segmentation system with improved accuracy.

B. Proposed Method

The structure of the system is similar to the one proposed previously in our works [17], [18]. A simple view of the system would be like the following:

- **Texture map generation step**
  - Texture map of input image is created based on texture descriptor.
- **Shape generation step**
  - Some new samples based on the training data are generated.
- **Segmentation step**
  1) Image containing the human subject is input.
  2) Labels are assigned to each pixel based on the generated mask from the shape 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 shape generation step using new parameters.

The rest of this section is dedicated to the explanation for each step depicted above.

C. Texture Map Generation

As mentioned before, in this work, we use the texture descriptor introduced by Zhou et al. [16]. This descriptor is relatively simple, thus easy to calculate and relatively fast.

The main idea behind the descriptor comes from the semi-local nature of the texture as mentioned in [16] and that the digital image can be considered as a result of sampling a smooth, differentiable manifold (usually referred to as Reimannian manifold) which makes us able to use the Beltrami representation [19] and differential framework to our favor.

(a) Subject with single textured area

(b) Subject with multiple textured areas

Fig. 4. Example of texture feature for an input image.

The color image can be depicted as the following Beltrami representation:

$$X(x,y) \rightarrow (X_1 = x, X_2 = y, X_3 = R(x,y), \\ X_4 = G(x,y), X_5 = B(x,y)) \quad (1)$$

in which x and y are coordinates and $R(x,y)$, $G(x,y)$, $B(x,y)$ are color values at that coordinate. Since texture has a semi-local and repetitive nature, we can select a window in the image and observe the rate of changes on the manifold like

$$P_1(x,y) = \Big\{R(x + w_x, y + w_y); w_x, w_y \in \Big[-\frac{n-1}{2}, \frac{n-1}{2}\Big]\Big\}. \quad (2)$$

In the same manner, $P_2(x,y)$ and $P_3(x,y)$ would be selected in Green and Blue channels, respectively.

Manifold representation in Eq. 1 will then become:

$$X(x,y) \rightarrow (X_1 = x, X_2 = y, X_3 = P_1(x,y), \\ X_4 = P_2(x,y), X_5 = P_3(x,y)). \quad (3)$$

Since the texture is usually repeated in a region, by observing the rate of surface change on manifold, we can have an estimation about if a region has an embedded texture or not. For this, we have to calculate the determinant of tensor matrix of the manifold:

$$G_{xy} = \begin{pmatrix} 1 + \sum_{i=1}^{3}(\partial_x P_i(x,y))^2 & \sum_{i=1}^{3}\partial_x P_i(x,y)\partial_y P_i(x,y) \\ \sum_{i=1}^{3}\partial_x P_i(x,y)\partial_y P_i(x,y) & 1 + \sum_{i=1}^{3}(\partial_y P_i(x,y))^2 \end{pmatrix} \quad (4)$$

Using this, we can calculate the texture feature as

$$T = \exp\Big(-\frac{\det(G_{xy})}{\sigma^2}\Big) \quad (5)$$

the Gaussian kernel helps us to control the degree of details to appear in the resulted texture map.The result of using Eq. 5 to calculate the texture map will be like the one presented in Fig. 4.

D. Shape Generation

Here, we use the statistical shape models (SSM) method for generating new shapes to be used as templates for segmentation. SSM method was proposed by Cootes et al. [20]. The following is its a brief explanation.

For generating new shapes, first, we train system with contour of some training shapes like $\mathbf{x}_i = [x_1, y_1, \ldots, x_n, y_n]^T$

(a) Case of $\mathbf{b} = [b_1, 0, \ldots, 0]^T$ and changing values of b_1.

(b) Case of $\mathbf{b} = [0, b_2, 0, \ldots, 0]^T$ and changing values of b_2.

Fig. 5. Some samples generated with SSM.

and calculate the mean shape and the covariance matrix from them as

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i \ , \tag{6}$$

$$\boldsymbol{S} = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T. \tag{7}$$

By calculating eigenvalues (λ_i) and eigenvectors ($\mathbf{p}_i$) of the covariance matrix and selecting the important ones (the axes with most changes), we can easily generate new shapes with the following equation:

$$\mathbf{x}_{\text{new}} = \bar{\mathbf{x}} + \boldsymbol{P}\mathbf{b} \tag{8}$$

Where $\boldsymbol{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_t]$ is a matrix containing the selected eigenvectors as columns, and $\mathbf{b} = [b_1, \ldots, b_t]$ is a vector of weights.

A suitable limit for the weights is described in [20] as

$$-3\sqrt{\lambda_k} \leq b_k \leq 3\sqrt{\lambda_k},\ k \in [1, \ldots, t] \tag{9}$$

Generating new shapes would be as easy as just changing the weights, e.g. $\mathbf{b} = [b_1, 0, \ldots, 0]^T$. Figure 5 shows how some shapes are generated by changing the values of $\mathbf{b}$.

E. Segmentation

For segmentation, here, we use a modified version of the famous Grab-cut algorithm which incorporates texture information into the segmentation framework. For this, the system is changed to accept texture information as 4th channel of data in addition to the three channels of information usually input (Red, Green, and Blue). The 4th channel is calculated as mentioned in Section III-C and as depicted in Fig. 7.

(a) Generated mask

(b) Resulted trimap

Fig. 6. Converting a generated sample to a trimap of "Foreground" (light gray), "Probably foreground" (gray), and "Probably background" (dark gray).

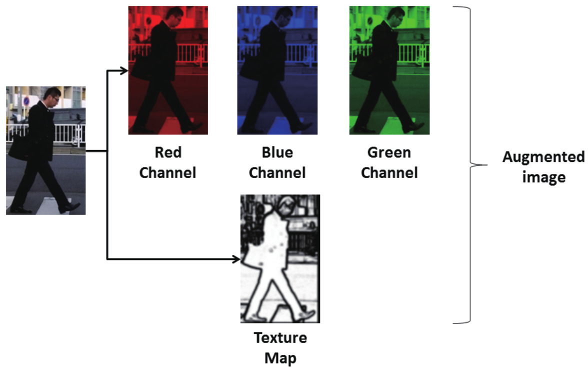


Fig. 7. Example of creating an augmented image.

There is also the distance penalty modification done in our previous work [17] to be mentioned.

Segmentation initialization is done by turning the generated shape into a trimap. A trimap can be defined as a pre-seeding map which tells the segmentation where to search for pixels to be separated in the image. Grab-cut uses four types of labels for each pixel "Foreground", "Background", "Probably Foreground", and "Probably Background". The first two labels cannot be changed after being set on a pixel while the other two will be changed during the process. For this work, we used the first three labels to create a trimap as depicted in Fig. 6. After generation, trimap will be input to the system along side the augmented image, which in Fig. 7, an example of it is presented. Based on these inputs, two GMMs are learned, one for foreground and one for background. These models are then utilized to differentiate between the foreground object and background in later steps.

F. Refinement

There are two stages of refinement here, local and global. in the local stage, after performing segmentation once based on the input image and the trimap, the result is compared with shapes generated by the SSM shape generator and the most similar shape is selected. The selected shape will then turn into a trimap and segmentation is performed again. This process will be repeated several times until one of the masks is selected repeatedly more than N_l times (i.e. the result of segmentation stabilizes).

In the global refinement stage, based on the parameters used to generate the best local shape and the previous best shape, a new set of N shapes is generated and the local refinement is repeated. Global refinement will be performed N_g times to achieve the best results. As mentioned in [17], these N_l and N_g are selected experimentally.

IV. Experimental Results

In this section, results of experiments for validating the proposed method are presented. For these experiments, 61

TABLE I
ACCURACY [%] COMPARISON BETWEEN COMPARATIVE METHODS.

	Dataset		
Method	Private dataset	PennFudan dataset	Average
Grab-cut	72.22	79.00	75.61
Watershed	79.33	80.74	80.04
One-cut	83.89	**85.76**	84.83
SSFSeg	85.43	80.54	82.99
Proposed	**87.07**	83.05	**85.06**

TABLE II
COMPARISON OF F1 MEASURE [%] BETWEEN COMPARATIVE METHODS.

	Dataset		
Method	Private dataset	Pennfudan dataset	Average
Grab-cut	67.21	75.57	71.89
Watershed	76.01	78.92	77.46
One-cut	80.41	**82.81**	**81.61**
SSFSeg	75.44	68.74	72.09
Proposed	**80.99**	76.77	78.88

samples for training the SSM model have been used. At each stage of sample generation, we created $N = 50$ samples from which, one was selected randomly for segmentation (the mean shape was used for the first segmentation). As for the number of times for local and global refinements, $N_l = 4$ and $N_g = 3$ were selected in this experiment, respectively.

Two datasets have been used for our experiments. The first dataset was 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 were taken with an in-vehicle camera and were color images with different sizes. The images were all taken during day time.

The second data set was derived from "Penn-Fudan database for pedestrian detection and segmentation" introduced by Wang et al. [21]. It contains 230 image with different sizes and all taken during day time.

The comparison was done between the original Grab-cut segmentation [5], Watershed [22], One-cut [11], SSFSeg [17], and the proposed method. The average segmentation accuracy and F1 measure was calculated for each method as depicted in Tables I and II.

For Grab-cut and Watershed segmentation methods, the code provided by the OpenCV open source library [23], and for One-cut Segmentation, the code provided by Tang el al. [24] were used.

As for the accuracy we used:

$$\text{Accuracy } (\%) = \frac{\text{TN} + \text{TP}}{\text{TN+TP+FP+FN}} \times 100 \tag{10}$$

And for F_1 measure, we used:

$$F_1 \ (\%) = \frac{2\text{TP}}{2\text{TP} + \text{FN+FP}} \times 100 \tag{11}$$

in which TP, TN, FP and FN show respectively the number of foreground pixels segmented correctly, the number of background pixels segmented correctly, the number of foreground pixels segmented as background, and the number of background pixels segmented as foreground. Figures 8 and 9 show the segmentation results by the proposed system and its comparison to other methods. As it can be seen, the system performs with higher accuracy compared to conventional methods. Tables I and II also present the average accuracy and F1 measure which also proves that the proposed method

(a) (b) (c) (d) (e)

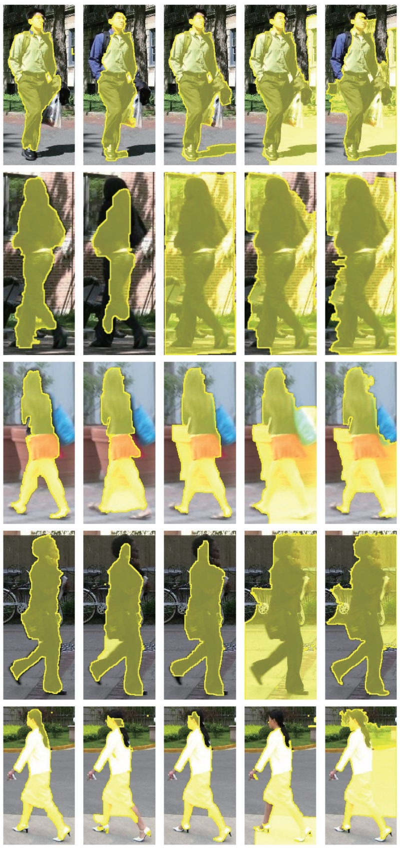
Fig. 8. Comparison between segmentation methods (Private Dataset). (a) Proposed method, (b) SSFSeg, (c) One-cut, (d) Grab-cut, (e) Watershed.

have better performance on accuracy while improving the F1 measure for SSFSeg [17] by 6%.

V. CONCLUSIONS

In this paper, we presented a method that can automatically segment pedestrians in images. The system incorporates texture alongside the color feature to improve the segmentation results. Our experiments show that using texture can help to improve the segmentation accuracy especially in cases where the color information is not enough to segment the desired object correctly.

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 work. Also, the method explained in this paper expects the output of a human detector algorithm



(a) (b) (c) (d) (e)

Fig. 9. Comparison between segmentation methods (PennFudan Dataset). (a) Proposed method, (b) SSFSeg, (c) One-cut, (d) Grab-cut, (e) Watershed.

(e.x. [25] and [26]) 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.

As for future work, we would like to:

- Make a more complete training dataset for the SSM generation step which includes more variations in the model.
- Make the system capable of coping with occlusions.
- 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 was developed based on the MIST library (`http://mist.murase.m.is.nagoya-u.ac.jp/`).

References

[1] 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, pp. 105–112, 2001.

[2] 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, pp. 18–25, 2005.

[3] 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.

[4] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," in *Proc. 22nd ACM Annual Conf. on Computer Graphics and Interactive Techniques*, pp. 191–198, 1995.

[5] 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.

[6] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof, "Tvseg—interactive total variation based image segmentation," in *Proc. British Machine Vision Conf.*, vol. 31, pp. 44–46, 2008.

[7] G. Varun, 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*, pp. 3129–3136, 2010.

[8] 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 Workshops*, pp. 1127–1133, 2011.

[9] 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.

[10] B. Peng and O. Veksler, "Parameter selection for graph cut based image segmentation.," in *Proc. British Machine Vision Conf.*, vol. 32, pp. 42–44, 2008.

[11] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov, "Grabcut in one cut," in *Proc. 2013 IEEE Int. Conf. on Computer Vision*, pp. 1769–1776, 2013.

[12] 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*, pp. 590–597, IEEE, 2012.

[13] 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*, pp. 628–635, 2013.

[14] 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.

[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] 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.

[17] E. Pourjam, H. Murase, D. Deguchi, and I. Ide, "Human subject segmentation based on grab-cut and statistical shape models," *IEEJ Trans. on Electronics, Information and Systems*, vol. 135, no. 8, pp. 1000–1008, 2015.

[18] 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.

[19] 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.

[20] 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.

[21] L. Wang, J. Shi, G. Song, and I.-F. Shen, "Object detection combining recognition and segmentation," in *Proc. 2009 Asian Conf. on Computer Vision*, pp. 189–199, 2007.

[22] S. Beucher and F. Meyer, "The morphological approach to segmentation: the watershed transformation," *Optical Engineering*, vol. 34, pp. 433–481, 1992.

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

[24] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov, "One cut with user seeds." , http://www.csd.uwo.ca/~ygorelic/downloads.html [accessed April-2015].

[25] 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*, vol. 1, pp. 886–893, 2005.

[26] 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*, vol. 1, pp. I–511—I–518, IEEE, 2001.