Research Paper

Live Video Segmentation in Dynamic Backgrounds Using Thermal Vision

VIET-QUOC PHAM,^{†1} KEITA TAKAHASHI^{†2}
and TAKESHI NAEMURA^{†1}

In this paper we describe a new technique for live video segmentation of human regions from dynamic backgrounds. Correct segmentations are produced in real-time even in severe background changes caused by camera movement and illumination changes. There are three key contributions. The first contribution is the employing of the thermal cue which proves to be very effective when fused with color. Second, we propose a new speed-up GraphCut algorithm by combining with the Bayesian estimation. The third contribution is a novel online learning method using cumulative histograms. The segmentation accuracy and speed are quite capable of the live video segmentation purpose.

1. Introduction

This paper addresses the problem of live video segmentation of human regions from dynamic backgrounds. Different from offline video segmentation that can use the whole video as the reference for the segmentation, live video segmentation does not employ future information, so that the estimation should be based only on the past frames. A prime application is live background substitution in broadcasting and teleconferencing. This demands layer separation with high quality and computational efficiency sufficient to attain live streaming speed. Besides, it must deal with dynamic background changes caused by several factors such as camera movement, illumination changes, etc.. However, by using only monocular video, even most effective recent researches^{1),2)} can at most deal with small camera shakes. Therefore, segmentation with more severe background changes like camera movement requires more effective cues besides color and motion.

Fig. 1 Camera system for thermo-key.

Fig. 2 Captured sequences from the thermo-key system: first row- color sequence, second row- thermal sequence.

An impressive decrease in both size and price of thermal vision cameras promises their wide applications in both researches and daily life. Aiming to achieve live video segmentation with high accuracy, Yasuda, et. al.³⁾ proposed a unique system named “thermo-key” which combines a color camera and a thermal vision camera. We have developed the same system as Yasuda’s one with a higher capturing speed. Both cameras share the same optical axis using an IR mirror as shown in **Fig. 1**, so that each pixel on the combined image has a thermal value as well as a color value. Sample sequences captured from this system are shown in **Fig. 2**. In this paper, we will prove that by using this thermal cue, we can attain more accurate segmentation results than Ref.1) even when using a simpler probabilistic segmentation model.

We propose three main contributions in this paper. The first one is the em-

^{†1} Graduate School of Information Science and Technology, The University of Tokyo

^{†2} IRT Research Initiative, The University of Tokyo

ploying of the thermal cue which proves to be very effective when fused with color. Because every person has his/her body temperature, which is often different from the surrounding environment temperature, thermal cue is a valuable feature to detect the human region for the segmentation task. Second, we propose a new speed-up GraphCut algorithm by combining with the Bayesian estimation. The advantage of GraphCut is that it can simultaneously consider texture (color+thermo) information and edge (contrast) information in its optimization scheme. It has been adopted for interactive image segmentation^{4),5)} and automatic video segmentation^{1),2)}. Our method aims to realize accurate human-region segmentation with lower complexity than Ref. 1) by combining the Bayesian estimation approach with a GraphCut algorithm. The third contribution is a novel online learning method using cumulative histograms which can be deal with background changes.

In this paper, previous works in video segmentation are described in Section 2. Section 3 introduces our proposed algorithm. Experiments and conclusions are presented in Section 4 and 5.

2. Related Works

Classical image segmentation tools use either texture (color) information, e.g. Magic Wand⁶⁾, or edge (contrast) information, e.g., Intelligent Scissors⁷⁾. Recently, an approach based on optimization by GraphCut has been developed which successfully combines both types of information. Boykov, et al.⁴⁾ and Rother, et al.⁵⁾ introduced powerful optimization techniques using GraphCut for interactive segmentation. However, although color/contrast cues have been demonstrated to be very effective for interactive segmentation, they alone are still insufficient for fully automatic segmentation. This suggests a need for a robust approach that fuses a variety of cues, for example depth, motion, and so on.

In Ref. 2), segmentation for static background was performed by combining background subtraction with the GraphCut optimization. Criminisi, et al.¹⁾ and Kolmogorov, et al.⁸⁾ described algorithms capable of real-time segmentation of foreground from background layers in video sequences, in which color, contrast and another key feature were fused to infer layers. The key feature used in Ref. 1) was motion, so that their algorithm could be applied to monocular videos, while

Kolmogorov, et al.⁸⁾ employed the depth feature obtained from stereo videos.

As stated before, the above researches can at most deal with small camera shakes. Segmentation with more severe background changes caused by camera movement requires more effective cues besides color and motion. The purpose of this research is to fuse color, contrast with *temperature* to realize real-time segmentation. Our work is an extension as well as an improvement of the previous work³⁾ which was based on a simple threshold processing to the thermal images (see Fig. 9).

3. Segmentation Algorithm

3.1 Assumption

The only constraint for our live video segmentation system is the requiring of background of the first frame. This simple assumption is reasonable because in most filming works, background is kept fixed before shooting. This constraint helps us to quickly detect the foreground region from background in the first frame by using a simple background subtraction method. We must notice that the background subtraction is applied for *only* the first frame. After starting capturing, the camera can move freely, making background changes, and therefore, the background subtraction will be no more in use.

The background subtraction can be performed on both color and thermal channels, however, only thermal channel is enough to produce rather good results. Although it sometimes does not work well due to the closed temperatures in some places between foreground and background, but by applying our robust segmentation algorithm, the segmentation errors appeared in the first frame will be removed in the next frames (see **Fig. 3**).

Fig. 3 In the segmentation result of the first frame, some errors appear because some parts of the object person have temperatures closed to the background temperature. However, these errors disappear in the next frames thanks to our robust segmentation method.

Some researches like Ref. 1) did not require background for the first frame, but they used the motion properties instead. When the object moves while background is kept static, the moving parts are detected and segmented. Much time was needed until the whole object is segmented (your hand will not be segmented until you move it). Furthermore, offline learning of motion properties was needed in some cases. For the live video segmentation purpose, it is clearly less practical than our assumption.

The segmentation algorithm for the next frames is stated in the successive sections.

3.2 Algorithm Overview

Figure 4 illustrates the entire algorithm. Our algorithm takes the color and thermal images captured by our thermo-key system as the inputs, and outputs the segmentation result (a binary mask) of the foreground region.

The first originality of our algorithm is the employing of the thermal feature combining with the color feature. Second, we propose a probabilistic model combining the Bayesian estimation approach with a GraphCut algorithm. Most GraphCut-based segmentation methods including Ref. 1) create the graph model for all image pixels. Such methods cost time because the calculation cost of the GraphCut algorithm relies mostly on the number of the graph nodes. Our contribution is that before applying GraphCut, we perform a pre-GraphCut step, or a trimap making step to reduce the unknown region (i.e., the region around the object boundary that is still unknown to be of the background or the foreground classes). As a result, we can speed up the GraphCut process. The third originality

Fig. 4 Demonstration for the two-step segmentation algorithm.

is a novel online learning method using cumulative histograms which can deal with background changes. These two steps of segmentation and online learning method are stated in the next sections.

3.3 Step 1- Pre-GraphCut Step

The first step calculates the probability of each pixel to be in foreground based on the Bayesian estimation. This calculation can be performed in the linear time. After that, the results are used to make the trimap.

For details, let C_i^t and T_i^t be the color and thermal values of the i^{th} pixel on the combined image at period t captured by the thermo-key system. Based on the Bayes' formula, the probability of this pixel to be in foreground is given as

$$P^t(F|C_i, T_i) = \frac{P^t(C_i, T_i|F)P_i^t(F)}{P^t(C_i, T_i|F)P_i^t(F) + P^t(C_i, T_i|B)P_i^t(B)} \quad (1)$$

where $P(*)$ represents the probability of “*”. F and B are foreground and background respectively, into which each pixel should be classified. The posterior probability, $P^t(F|C_i, T_i)$, is represented as the product of the likelihood $P^t(C_i, T_i|F)$, and the prior probability, $P_i^t(F)$. The likelihoods $P^t(C_i, T_i|F)$ and $P^t(C_i, T_i|B)$ are calculated from the foreground and background histograms which were constructed from the previous segmentation results. These histograms are used to discriminate foreground pixels from background ones, and are initialized from the segmentation result of the first frame which was attained by background subtraction (see Section 3.1). Since there are some correlations between color and thermal information, we should use a 4-D cumulative histogram, where 3 dimensions are assigned for YUV color channels, and another dimension is for the thermal channel. Based on empirical study, we have found that modeling the correlation of color and thermal cues in this way gave much better results than the case when the two cues are independently modeled with two separate histograms and combined by applying product rule. The meaning of cumulative histogram is explained in Section 3.5.

The prior probability is calculated from the previous image frame since successive frames in the temporal domain would have strong correlations. To be more precise, let α^{t-1} be the 2-D mask of the segmentation result (taking 255 for F and 0 for B) at period $t-1$, and M^t be the smoothing map of α^{t-1} which is defined as follows

Fig. 5 The smoothing map is created from the segmentation result of the previous frame.

$$M^t = G_{7 \times 7}(\text{Resize}_{\times 10}(G_{3 \times 3}(\text{Resize}_{\times 1/10}(\alpha^{t-1}))))/255 \quad (2)$$

where G is the Gaussian filter and Resize is the scaling transformation. Then the prior probability $P_i^t(x)$ can be inferred from M^t

$$P_i^t(x) = \begin{cases} M_i^t & (x = F) \\ 1 - M_i^t & (x = B) \end{cases} \quad (3)$$

An image of a smoothing map is shown in **Fig. 5**.

3.4 Step 2- GraphCut Optimization

The set of posterior probabilities $P^t(F|C_i, T_i)$ over all image pixels i forms the probability map I_{prob} . We create the trimap, $Tr(i)$, which takes one of the three values $\{F$ (foreground), B (background), and U (unknown) $\}$, based on the value of $I_{prob}(i)$

$$\begin{cases} \text{if}(I_{prob}(i) < \epsilon) : & Tr(i) = B \\ \text{if}(I_{prob}(i) > 1 - \epsilon) : & Tr(i) = F \\ \text{otherwise} : & Tr(i) = U \end{cases}$$

where ϵ is a small real value. Then, a filter is applied to the foreground and background regions of the trimap to remove noises. Since such clusters of noise pixels seem to be relatively small in size, this filter operation can be achieved in linear time. An image of a trimap is shown in **Fig. 6**.

Before getting further into our proposed method, it is better to explain briefly about the GraphCut optimization algorithm⁴⁾. Let Seg be a segmentation of the image where $Seg(i)$ takes F (foreground) or B (background) for the i^{th} pixel.

Fig. 6 A trimap is a classification of an image into three regions: foreground (white), background (black) and unknown (gray).

We define the energy function by

$$E(Seg) = \text{Data}(Seg) + \lambda \text{Sth}(Seg) \quad (4)$$

where the data term $\text{Data}(Seg)$ evaluates the pixel-wise costs, and the smoothing term $\text{Sth}(Seg)$ evaluates the inter-pixel costs. λ is a weighting coefficient. This function can be minimized by the GraphCut algorithm, and the optimization solution should produce a good segmentation because it keeps the balance between the region property and the boundary property of the segments.

In our method, the above filtered trimap is set as the initial values for the GraphCut optimization. Pixels in the foreground and background regions of the trimap are used as the “seeds” for image segmentation, i.e., in the final segmentation result: $Seg(i) = Tr(i) \forall i$ satisfying $Tr(i) \in \{F, B\}$. In some interactive segmentation method like 4), these “seeds” are user-supplied, while in our method, they are generated automatically. The GraphCut process is needed to perform only on the unknown region of the trimap. As we can see from Fig. 6, the unknown region is much smaller than the whole image. Because the GraphCut process is applied only on the unknown region, its calculation speed can increase considerably.

We define the data term as the weighted sum of logs of the likelihood $P(C_i, T_i | Seg(i))$, and the probability map given by the first step $I_{prob}(i)$

$$\begin{aligned} \text{Data}(\text{Seg}) = & (1 - \mu) \sum_i -\log(P(C_i, T_i | \text{Seg}(i))) \\ & + \mu \sum_i -\log(I_{\text{prob}}(i)) \end{aligned} \quad (5)$$

where the likelihood can be obtained from the cumulative histograms (Section 3.5). The data term measures the agreement between the segmentation labels and the observed data. In some interactive segmentation methods like Ref. 4), only color appearance was used to build this term. This property is not enough to discriminate background from foreground due to some common colors appearing in both foreground and background histograms. In our method, in addition to color-thermal appearance (the first term of Eq. (5)), we employ a new property called spatial-temporal coherence (the second term of Eq. (5)) to build the data term. This coherence can be found easily by reusing the probability map $I_{\text{prob}}(i)$, which combines the color-thermal coherence and the temporal coherence in Eq. (1). This is another different point of our proposed method from the others.

To set the smoothing term, we use an ad-hoc function which was used by most other methods

$$\text{Sth}(\text{Seg}) = \sum_{(i,j) \in N} [\text{Seg}(i) \neq \text{Seg}(j)] \frac{e^{-\|C_i - C_j\|^2 / (2\sigma^2)}}{\|\text{dist}(i, j)\|} \quad (6)$$

where N represents all neighborhood pixel pairs, and $\text{dist}(i, j)$ denotes Euclidean distance between i^{th} and j^{th} pixels. σ can be estimated as “camera noise”. This function imposes larger cost to the discontinuous regions with little color variations. Our implementation of the GraphCut algorithm uses the minimum-cut algorithm from Ref. 9).

3.5 Cumulative Histogram

The main problem of segmentation from dynamic backgrounds is how to deal with the circumstance where the textures of foreground and background are changing along the time (see **Fig. 7**). The conventional solution to updating foreground/background likelihoods is adding up histograms of pixel values, which are based on data from the segmented foreground and background in each previous frame, adaptively over successive frames. However, based on our experiment re-

Fig. 7 When the camera moves, textures of background change noticeably. A new object (a laptop PC) appears in the second scene, and the illuminations in the first and third scenes are clearly different.

sults, this solution did not work well for long video sequences. The reason is that segmentation results always include errors, and if we update all bins (a bin is a certain range of 4D vectors expressing the color and thermal values), these errors will propagate, and as the sequence, both foreground and background histograms will converge to each others, making the discrimination between foreground and background becomes impossible.

In this section, we propose an effective method for updating cumulative histograms. First, we create two normalized temporary histograms, one for foreground and one for background, based on data from the segmented foreground and background in the previous frame. Cumulative foreground and background histograms are then updated from the two temporary histograms as follows: *only zero bins*, i.e., bins which have the current values of zeros in the *both* cumulative histograms, are updated with the correspondent values from the temporary histograms. Finally, the two cumulative histograms are normalized. Using this method, we can ensure that, newly appearing color-thermal values will be updated to cumulative histograms, and every bin is updated at most once, keeping the errors from propagation. The number of bins was chosen carefully based on various experiments. Based on our empirical study, we found that 16 bins per each channel (three YUV channels and one Thermal channel) works well. Detail evaluation of our proposed method is stated in Section 4.1. A demonstration is

Fig. 8 Our proposed method for updating cumulative histograms.

described in **Fig. 8**. To make the demonstration easy to understand, we used 1D histograms instead of the 4D ones.

4. Experimental Results

In our experiment, we captured two video sequences named “Video Chatting” and “Weather Forecast”. Each frame in each sequence is a pair of a 24-bit color image and a 8-bit thermal image (both are 320×240 pixels). In “Video Chatting” (**Fig. 10** (a)), a person who is the segmentation object was talking in front of the camera while the background was changing following the movement of the cameras. In “Weather Forecast” (**Fig. 10** (b)), the person was moving backward and forward while making some hand gestures during his presentation. The capturing system was slowly panned during the sequence to track the person’s position. Besides, some objects like monitors whose temperatures closed to the body temperature were located in the backgrounds, making the backgrounds more dynamic in both color and thermal channels. Therefore, the segmentation task became so difficult that a simple thresholding method based on the thermal information, which was employed in the earlier implementation of the thermo-key system³⁾, became inapplicable in such circumstances (see **Fig. 9**).

Four coefficients ($\epsilon, \lambda, \mu, \sigma$) in Section 3.4 were set as ($\epsilon = 1/32, \lambda = 20, \mu =$

Fig. 9 Results from the thresholding method applied to the 100th frame from sequences (a) “Video Chatting” and (b) “Weather Forecast”. From left to right: input color images, input thermal images, thresholding results. Some places in the background whose temperatures are closed to the body temperature were misclassified.

$0.75, \sigma = 27$).

4.1 Segmentation Quality

For each sequence, we captured the background image in advance before filming, then applied background subtraction on the first frame, to extract the foreground region from background. We then initialized the cumulative histograms with these segmented foreground and background data. Finally, we applied the proposed method stated in Section 3 to segment the next frames live and automatically. Segmentation results are shown in **Fig. 10**. The first and second rows show the input color and thermal sequences captured from the thermo-key system. The third row shows the segmentation results. On the last row, the segmentation results are superimposed on another background. The entire results are provided in the supplemental files.

To evaluate the results, we first made a comparison of three cases, in which our proposed method was performed 1) using only color feature, 2) using only thermal feature, and 3) fusing the two features. Second, to prove the effectiveness of our method, we also compared the performance of our method when only color feature was used (i.e., without fusing the thermal one) with one of the

Fig. 10 Segmentation results from two sequences (a) “Video Chatting” and (b) “Weather Forecast”. First row: input color sequence, second row: input thermal sequence, third row: output segmentation results, last row: background substitution results. Notice that the cameras were panned in both sequences.

most effective conventional methods- Bilayer Segmentation¹⁾. This method was implemented from the source code provided on the website^{*1}. The temporal prior and the motion likelihood required by this method were learned from the ground truths of the first 100 frames. To evaluate the quality of the segmentation results,

*1 <http://vision.caltech.edu/projects/yiw/FgBgSegmentation/>

Fig. 11 F-measures calculated over 370 frames from two sequences (a) “Video Chatting” and (b) “Weather Forecast”.

we used F-measure which is the harmonic mean of precision and recall

$$F = 2 \cdot (\text{precision} \cdot \text{recall}) / (\text{precision} + \text{recall}) \quad (7)$$

The hand-labeled ground truths were used to calculate the precision and recall. We must notice that these ground truths were only used for the evaluation and *not* employed in our algorithm at all. The graphs of F-measures calculated over 370 frames from the two sequences showed the comparison results (see **Fig. 11**).

In Fig. 11 (b), using only color information produced poor results because the background changed noticeably in both its texture and illumination; therefore, the information obtained from the thermal camera dominates the segmentation results. In contrast, for the sample in Fig. 11 (a), using only thermal information produced poor results because some objects like monitors whose temperatures closed to the body temperature were located in the background; therefore, in this case, the information obtained from the color camera dominated the segmentation results. In both cases, combining both color and thermal information gave the best results where the F-measures were always higher than 95%. Bi-layer Segmentation¹⁾ did not work for the sample in Fig. 11 (b) because its motion model did not adapt to severely changing backgrounds. Besides, because their algorithm needs to be trained by lots of hand-labeled segmentation data to build the temporal prior, color, and motion models, their method was not capable of the live video segmentation purpose.

In the end of this section, we evaluate the effectiveness of our proposed online learning method stated in Section 3.5. The cumulative foreground and background histograms are initialized using data from the segmented foreground and background in the first frame. We then compare the effectiveness of two updating methods: our proposed method which updates only zero bins, and the conventional method which updates all bin by adding up histograms adaptively over successive frames. In the conventional method, the newly added histogram, which is based on the segmentation data from the current frame, is weighted with a factor β . We varied this weighting factor with three values: 0 (not update at all), 1 (choose the new histogram as the cumulative histogram), and 1/2 (add the new histogram with proportion of 50%), then compare the segmentation quality with our proposed method. The sequence “Weather Forecast” was employed in this experiment and the comparison result is shown in **Fig. 12**. When $\beta = 0$, i.e. the cumulative histograms were not updated at all, the segmentation task failed due to the changes of background and foreground textures along the time. When $\beta = 0.5$ or 1, the segmentation task failed due to error propagation (see Section 3.5). Our proposed method is so effective that the segmentation quality remained high for a long time.

Fig. 12 Comparison between our proposed updating method and the conventional method with three different weighting factors.

4.2 Calculation Cost

Our experiments were performed on a PC with Intel Core2 Quad CPU, 2.40 GHz and 4 GB memory. The calculation time for segmentation was about 90 milliseconds for each frame. To prove the fact that initializing the foreground and background regions using the pre-GraphCut step (see Section 3.3) could speed up the GraphCut process, we also measured the calculation time for the case in which the foreground and background regions were not initialized, i.e., the GraphCut algorithm was applied over the entire image region. This calculation time was about 200–300 milliseconds for each frame which was much slower than the proposed method.

4.3 Live Video Segmentation

In the live video segmentation system, the delay times (the waiting time until the result is outputted) must be kept as short as possible. In our implementation, besides employing the speed-up GraphCut algorithm, we also applied parallel processing for the capturing phase and the segmentation processing phase. As a result, the delay time was decreased to less than 200 milliseconds (about two frames’ processing time). **Figure 13** shows our live video segmentation demonstration in which the segmentation results are displayed behind the person with a few frames’ delay.

Fig. 13 Live video segmentation demonstration. The segmentation results are displayed live in the behind projector screen.

4.4 Other Applications

In this section, we propose two other applications, besides background substitution, inferring from our segmentation system: matting and clothes synthesis.

The purpose of matting is to process fuzzy objects (like hair, feathers, etc.) near the object contour and produce correct fractional opacity channel for them. It is used as a post-processing of segmentation, for anti-aliased compositing the segmented object onto a new background. Here, we segment each frame of the input video into foreground and background, then apply Bayesian matting¹⁰⁾ to the region around the boundary. Some results from a sample sequence are shown in **Fig. 14**.

The second application called “clothes synthesis” aims to synthesize the clothes region with different colors and patterns automatically. After extracting the human region from the sequence, we employ the difference in temperature between the clothes and the skin, to extract the clothes region from the whole body region. Then we can make change to the color and the pattern of the clothes, as shown in **Fig. 15**.

5. Conclusions

In this paper, we have developed an effective human region segmentation

Fig. 14 Matting results from a sample sequence. The first row shows input color frames, the second row shows matting results. High matting quality was attained in spite of dynamic backgrounds caused by camera movement.

Fig. 15 Clothes synthesis application.

method for the live video segmentation system. The experimental results showed that our method is more effective than conventional methods for segmentation from dynamic backgrounds. There are three key contributions. The first contribution is the employing of the thermal feature. Second, we propose a new

speed-up GraphCut algorithm by combining with the Bayesian estimation. The third contribution is a novel online learning method using cumulative histograms. The segmentation accuracy and speed are quite capable of the live video segmentation purpose. Some practical applications using our segmentation system are also proposed in this paper. Our future work will be focused on further optimization of the algorithm.

References

- 1) Criminisi, A., Cross, G., Blake, A. and Kolmogorov, V.: Bilayer Segmentation of Live Video, *Proc. IEEE Computer Vision and Pattern Recognition*, Vol.1, pp.53–60 (2006).
- 2) Sun, J., Zhang, W., Tang, X. and Shum, H.: Background Cut, *Proc. European Conference on Computer Vision*, pp.628–641 (2006).
- 3) Yasuda, K., Naemura, T. and Harashima, H.: Thermo-key: human region segmentation from video, *IEEE Trans. Computer Graphics and Applications*, Vol.24, pp.26–30 (2004).
- 4) Boykov, Y. and Jolly, M.: Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images, *Proc. International Conference on Computer Vision*, Vol.I, pp.105–112 (2001).
- 5) Rother, C., Kolmogorov, V. and Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts, *Proc. SIGGRAPH*, Vol.23, pp.309–314 (2004).
- 6) Adobe Systems Inc.: *Adobe Photoshop User Guide* (2002).
- 7) Mortensen, E. and Barrett, W.: Intelligent scissors for image composition, *Proc. Computer graphics and interactive techniques*, pp.191–198 (1995).
- 8) Kolmogorov, V., Criminisi, A., Blake, A., Cross, G. and Rother, C.: Bi-layer segmentation of binocular stereo video, *Proc. IEEE Computer Vision and Pattern Recognition*, pp.407–414 (2005).
- 9) Boykov, Y. and Kolmogorov, V.: An Experimental Comparison of Min-cut/Max-flow Algorithms for Energy Minimization in Vision, *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp.1124–1137 (2004).
- 10) Chuang, Y.Y., Curless, B., Salesin, D.H. and Szeliski, R.: A Bayesian approach to digital matting, *Proc. IEEE Computer Vision and Pattern Recognition*, pp.264–271 (2001).

(Received March 30, 2009)

(Accepted December 21, 2009)

(Released March 11, 2010)

(Communicated by Fay Huang)

Viet-Quoc Pham received his B.S. and M.S. degrees in information and communication engineering from the University of Tokyo, Japan, in 2007 and 2009, respectively. He is currently a Ph.D. candidate in department of information and communication engineering, the University of Tokyo. His research interests include machine learning, image segmentation, object recognition and digital watermarking.

Keita Takahashi received his B.E., Master and Ph.D. degrees in information and communication engineering from the University of Tokyo, Japan, in 2001, 2003, and 2006, respectively. He is currently a Project Assistant Professor of IRT Research Initiative, the University of Tokyo. His research interests include 3D image production, object recognition, and video segmentation.

Takeshi Naemura received his BE, ME, and Ph.D. degrees in electronic engineering from the University of Tokyo in 1992, 1994, and 1997, respectively. He is currently an associate professor of information and communication engineering at the University of Tokyo. His research interests include image-based rendering, 3D image processing, human interface, mixed reality, and computational photography.