Hand Mouse System Using a Pre-defined Gesture for the Elimination of a TV Remote Controller

Kyung-Won Kim[1], Daehee Bae[2], Joonhwan Yi[2], and Seong-Jun Oh[1]

[1] Department of Computer and Commun. engineering, Korea University / Seoul, South Korea
{kyungwon, seongjun}@korea.ac.kr
[2] Department of Computer engineering, Kwangwoon University / Seoul, South Korea
{placebo1014@gmail.com, joonhwan.yi@kw.ac.kr}

* Corresponding Author: Seong-Jun Oh

Received July 14, 2012; Revised August 18, 2012; Accepted September 5, 2012; Published October 31, 2012

***Abstract*:** Many hand gesture recognition systems using advanced computer vision techniques to eliminate the need for a TV remote controller have been proposed. Nevertheless, some issues still remain, such as high computational complexity and insufficient information on the target object and background. Moreover, none of the proposed techniques consider how to enter the control mode of the system. This means that they may need a TV remote controller to enter the control mode. This paper proposes a hand mouse system using a pre-defined gesture with high background adaptability. By doing so, a remote controller to enter the control mode of the IPTV system can be eliminated.

***Keywords*:** Hand Mouse, Moving Object Detection, Object Tracking, Frame Voting

1. Introduction

Currently, the importance of the interaction between a human and computer is emphasized due to the increase in various digital devices. In particular, the user interface using human motion has come into the spotlight as a human-computer interaction (HCI). The user interface using human motion is used for a range of applications, such as augmented reality, video games and controlling Internet Protocol TeleVision (IPTV). Using the motion recognition system, digital devices, such as IPTV can be controlled without the need for a remote controller. Furthermore, a range of services that had previously been impossible will become achievable with the proposed system.

To control IPTV using the HCI systems, it is important to determine the target object that controls the device. A range of object detection and tracking techniques have been used to achieve this. Almost all object detection algorithms use the color and/or the shape of the target object. Skin color information is commonly used to detect the target object because the hand is the most common target object [1]. On the other hand, object detection algorithms based on skin color are affected by the individual or racial difference of skin color and illumination. Although they are not based on skin color, all specific color-based object detection algorithms have the same weakness. The object detection algorithms based on the shape of the target object [2] are influenced less by the color or illumination of the target object, but cannot distinguish what is the target object when many hands are observed. The template matching method [3] uses both color and shape, but cannot overcome the weakness. The observed hands can be distinguished using face recognition [4] and pose estimation [5] techniques, but it requires information on the user's face.

To overcome these problems, this paper proposes a hand mouse system using a pre-defined gesture. The proposed system detects all moving objects in a video frame, and the observed objects are linked to motion sequences. If a motion sequence is detected as a pre-defined gesture, the object of the motion sequence is defined by the target object. The target object can be detected without additional information, such as shape and color. On the other hand, the proposed algorithm is based on movement, making the target object's boundary difficult to detect accurately. Unfortunately, when target object detection is inaccurate, the performance of the

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2011-0009454 & No. 2011-0025385 & No. 2010-0028204)

object tracking algorithm is degraded because an inaccurate region of the target object can include a high proportion of background regions. To supplement this weakness, the Adaptive Background model for the Camshift (ABCshift, [14]) algorithm was used to track the target object. The ABCshift algorithm has good adaptability to the background. The proposed hand mouse system can then recognize a hand without additional information with good tracking performance.

The remainder of the paper is organized as follows. Section II explains the overall structure of the proposed system, and Section III explains the proposed system. Section IV shows the simulation result. Finally, the conclusions are reported in section V.

2. Overall Structure

The proposed algorithm consists of three modes: observe mode, learning mode and control mode. Fig. 1 shows the flow of the proposed system in three modes. The aim of the algorithm in observe mode is to detect the target object. The target object is determined by the hand that shows the pre-defined gesture. The algorithm in learning mode prepares to track the target object, where they collect information on the target object. The algorithm in control mode tracks the target object to provide a range of services for the user.

In observe mode, the proposed algorithm operates as follows. First, a pixel-based moving object detection algorithm is used. The detected moving pixels are converted to object units using the clustering algorithm. After moving objects are detected, the motion sequences of the moving objects are generated over many video frames. The sequence detection algorithm is used to determine if each sequence is a pre-defined gesture. If one of the sequences is a pre-defined gesture, the object is then determined by the target object and the system turns over to learning mode.

In learning mode, the size of the target object is determined by machine learning. During size learning, if the size of the target object is out of the permissible range, learning fails and returns to observe mode. If the size learning is complete, the system turns over to control mode.

In control mode, the ABCshift algorithm is used to track the target object. During tracking, the user can use all IPTV services, such as web surfing. On the other hand, this paper does not deal with applications but only with the tracking algorithm. If the command gesture for the finish is detected, the system returns to observe mode.

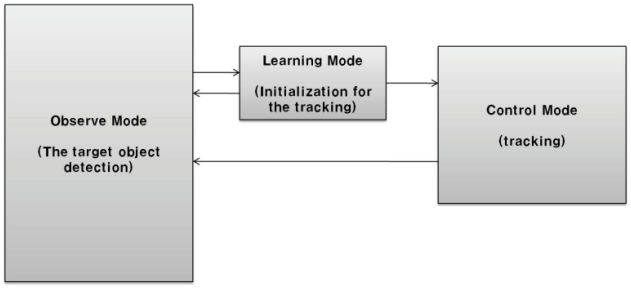

Fig. 1. The flow of the proposed system with three modes.

3. The Proposed Algorithm

3.1 Observe Mode

Fig. 2 shows the flow of the proposed algorithm in observe mode. Observe mode detects the target object using a pre-defined gesture. The proposed algorithm follows the following four steps: pixel based moving object detection, noise elimination, clustering and gesture detection.

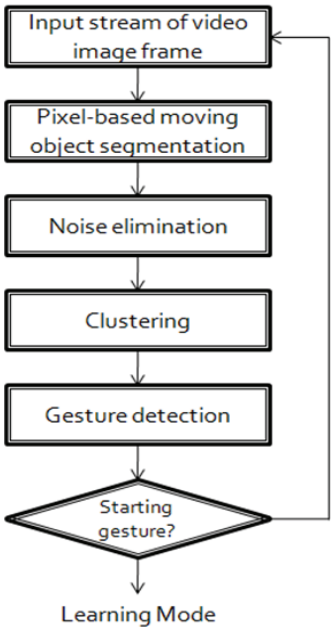

Fig. 2. The flow chart of the proposed algorithm in observe mode.

3.1.1 Pixel-Based Moving Object Detection

In the proposed system, the frame voting algorithm [6] is used to detect moving objects. Many pixel-based moving object detection algorithms are explained in reference [7]. The frame voting algorithm is based on the difference in images between the present frame and previous frames. The advantages of the algorithms based on the differences in images are the low computational complexity and high sensitivity to temporal changes. Although they are weak in terms of noise and afterimages, they are good for situations when there is no background information. The frame voting algorithm complements the weaknesses of the afterimages. Using the frame voting algorithm, improved performances can be achieved with low computational complexity.

The frame voting algorithm uses the temporal differences in color over a set of multiple recent frames. In this paper, the recent eight frames were used. The frame

voting formula of the moving object in the n[th] frame can be expressed as follows:

$$\mathrm{FV}_n(x,y) = \sum_{k=n-7}^{n-1} \mathrm{CNT}\left(\left\|I_n(x,y) - I_k(x,y)\right\|^2\right) \quad (1)$$

where

$$\mathrm{CNT}(a) = \begin{cases} 1 & \text{if } a > \tau \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

and I_k is an image matrix of time k, where $I_k(x, y)$ is the RGB color value of (x, y) coordinate in the k^{th} image matrix. τ is an empirical threshold.

The frame voting algorithm counts up every time the temporal difference between the present frame and each previous frame is larger than τ. If the total counted number, FV(x, y), is larger than 3, the pixel of (x, y) is detected as the pixel on a moving object. Fig. 3 shows an example of the result of frame voting. In the simple difference image (left), the hole of the arm by the afterimage effect is clearly observed, but it is removed in the result image of the frame voting algorithm. The frame voting algorithm has strong resistance to the afterimage effect, but the afterimage effect still exists if an observed object moves too slowly. In addition, the frame voting algorithm is so sensitive that small human vibrations can be a detrimental factor. Owing to these two weaknesses, the noise elimination algorithm should be done after the frame voting algorithm.

Fig. 3. Comparisons of pixel-based moving object detection algorithms when n=14. The left image is the result of the simple difference image, and the right image is the result of the frame voting algorithm.

3.1.2 Noise Elimination

Noise caused by vibrations after pixel-based moving object detection can be serious because it is based on temporal differences. To overcome this, the erosion-dilation algorithm is used to eliminate noise. The erosion-dilation algorithm is not a powerful noise elimination algorithm, but has low computational complexity and is sufficient to eliminate the noise caused by vibrations.

The erosion-dilation algorithm performs the erosion operation first followed by the dilation operation. The erosion and dilation operations are the fundamental operations in morphological image processing, which are explained in reference [8]. The noise is eliminated by the erosion operation, but the detected object becomes small. Therefore, its size is restored by the dilation operation.

3.1.3 Clustering

To detect a pre-defined gesture, each set of moving pixels should be classified by a clustering algorithm and converted to object units. The K-mean algorithm [9] and Gaussian mixture model (GMM) [10] are well-known algorithms for clustering. On the other hand, to implement them, it is important to know how many moving objects exist. The density-based spatial clustering of applications with a noise (DBSCAN) algorithm [11] does not need to know about moving objects and achieves good performance. Therefore, it is used in the proposed system. On the other hand, it has high computational complexity, so the DBSCAN algorithm based on the sub-pixel is used. A sub-pixel consists of 5x5 pixels, and the sub-pixel generation process can be substituted by down-scaling.

The DBSCAN algorithm uses two global parameters, Eps and MinPts. Eps is the maximum distance between pixels when comprised of a single cluster. MinPts is the minimum number of points when being comprised of a single cluster. The DBSCAN algorithm defines the three types of points: core point, border point and noise point. The core points are the pixels with more neighboring pixels than MinPts within Eps. The border points are not the core points, but the neighboring pixels of the core points. The noise points are the remaining pixels.

The DBSCAN algorithm is operated in four steps. In the first step, all points are classified as the core points, border points and noise points. In the second step, the noise points are removed. In the third step, all core points around Eps are classified as a single cluster. In the final step, each border point is classified as the cluster of the nearest core point. Subsequently, all pixels, except for the noise pixels, are converted to an object unit. As the purpose was to detect a hand gesture, a wrist elimination algorithm was used after the DBSCAN algorithm. If the height of the bounding quadrangle of each object is longer than its width, then the lower part of the object is removed.

3.1.4 Gesture Detection

The sequences of the moving objects should be generated for gesture detection. The Viterbi algorithm [12] was used to generate the sequences. The Viterbi algorithm is the sequence detection algorithm. On the other hand, in this paper, it was used to estimate the object motions. The Viterbi algorithm is optimal if the sequence follows the Markov properties. Using the equation of the optimality of Viterbi algorithm, the sequence $\{q_1^* q_2^* \ldots q_t^*\}$ is generated by following:

$$\left\{q_1^* q_2^* \ldots q_t^*\right\} = \max_{q_1 q_2 \ldots q_t} \left\{P(q_t) \prod_{k=1}^{t-1} P(q_k \mid q_{k+1})\right\}. \quad (3)$$

According to (3), each object selects the object with the highest conditional probability $P(q_k|q_{k+1})$, and the optimal sequences are generated by linking them. In this study, it was assumed that the conditional probability is highest when the distance between the mean locations is shortest. The mean location (m_{10}, m_{01}) was obtained by the first moments, which were obtained by

$$m_{ab} = \sum I_n(x,y) \cdot x^a x^b \quad (4)$$

Fig. 4 shows the optimal sequences generated by the Viterbi algorithm. Each object selects one of the next objects on its right, and the sequences of each object in the present frame (the farthest left) are then generated. In Fig. 4 the sequence of the red dotted line {000121} is the optimal sequence for object #0 in the present frame. In the figure, three sequences {000121}, {111000} and {222121} are detected, and all motion sequences of the objects in the present frame are then obtained.

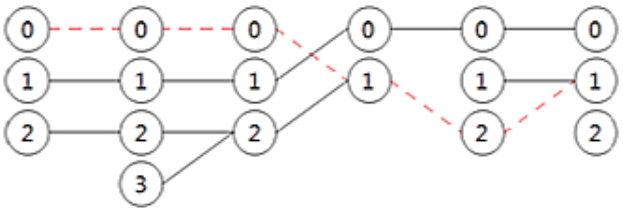

Fig. 4. The optimal sequences generated by the Viterbi algorithm. The number of moving objects can be different according to the frame.

The assumption was that the motion of a pre-defined gesture is shaking from side to side. By this assumption, whether or not each motion sequence is a pre-defined gesture is detected by the changes in m_{10}, and m_{01} is not used because it is affected significantly by wrist motion. If direction of changes of m_{10} turns the direction more than three times within 15 frames, the motion sequence is detected as a pre-defined gesture, and the object is determined by the target object. On the other hand, there is no need to select the sequence if one of the changes in m_{10} is too large. After the target object is determined, the proposed system changes to learning mode.

3.2 Learning Mode

The learning mode begins when the target object is determined in observe mode. In learning mode, the size of the region of interest (ROI) is determined based on the assumption that the shape of ROI is a circle. A circle is one of the simplest shapes that can include the entire hand. The size of the ROI is then determined by the radius of the ROI, and is defined as the average radius of bounding circles of the moving objects within 5 frames. If at least one of the radii of the bounding circles is out of the permissible range, size learning fails and the system returns to observe mode. If the size learning succeeds, the proposed system changes to control mode.

3.3 Tracking Mode

The ABCshift [14] algorithm is used to track the target object. The ABCshift algorithm has high background adaptability. The ABCshift algorithm is based on the mean-shift tracking algorithm. The mean-shift tracking algorithm is good for when the shape of the target object is changed dynamically as in a hand gesture. Furthermore, the mean-shift algorithm has low computational complexity.

In the tracking algorithms based on the mean-shift algorithm, the continuously adaptive mean-shift (CAMshift) algorithm [13] is commonly used, but does not consider the background. Therefore, the ABCshift algorithm [14] is proposed. The ABCshift algorithm has greater adaptability than the CAMshift algorithm because of its ability to allocate weights more efficiently in a Bayesian sense by considering the background.

3.3.1 Mean-Shift Tracking Algorithm

The mean-shift algorithm is the optimization algorithm that searches the local maximum. In tracking, the mean-shift algorithm searches the kernel with the largest sum of weights of pixels near the ROI of the previous frame. The weights of the pixels are determined by their color, and the weight function is explained in the next sub-section.

The mean-shift algorithm follows the following five iterative steps. In step 1, the size of the search window is set. In step 2, the center of the search window is set by the center of the ROI. In step 3, the mean location (m_{10}, m_{01}) of the search window is calculated. The mean location is obtained by the first moment of weights, which is determined by the feature-color. In step 4, the center of the search window is set by the mean location obtained in step 3. In step 5, steps 3 and 4 are repeated until convergence is achieved. If the mean location converges, the mean location is set as the center of a new ROI of the present frame.

3.3.2 Weight Function of the ABCshift Algorithm

Fig. 5 shows the ROI and adjacent background region (ABR). The red circle in Fig. 5 is the ROI, which is detected by the proposed object detection algorithm. The ABR is defined as the neighborhood region of the ROI, and in the present case, is a donut with the same center point of the ROI but double the radius. The proposed algorithm detects the location of the target object but the ROI also includes a substantial portion of the white wall. The white color pixels provide faulty information regarding the color of the target object. This is critical to the color-based tracking algorithm, but the ABCshift algorithm can supplement the weakness using a modified weight function.

In the ABCshift algorithm, the probability that a color c is in ROI is used as the weight function. To obtain the weight function, the conditional color probabilities are defined as follows:

$$P(I(x,y) = c \mid \mathrm{ROI}) = \frac{N_{\mathrm{ROI}}(c)}{N_{\mathrm{ROI}}} \quad (5)$$

and

$$P(I(x,y) = c \mid \mathrm{BG}) = \frac{N_{\mathrm{ROI}}(c)}{N_{\mathrm{BG}}} \quad (6)$$

Fig. 5. The region of interest (ROI, red circle) and adjacent background region (ABR, green donut).

where $I(x, y)$ is the color of the pixel of coordinate (x, y), and $N_{ROI}(c)$ and $N_{BG}(c)$ are the numbers of pixels in color c in the ROI and background region, respectively, and N_{ROI} and N_{BG} are the total number of the pixels in ROI and background, respectively. In reference [14], the background region is defined as the entire region except for the ROI, but in the present case, it is defined as the ABR. Because all the tracking algorithms search the target object around the previous location, the remaining region except for the ABR is unnecessary. In addition, the color distribution of ABR is more important than the color distribution of the entire background region.

Using the Bayes' rule, the conditional probability (5) can be expressed as

$$P(\mathrm{ROI} \mid I(x,y)=c) = \frac{P(I(x,y)=c \mid \mathrm{ROI})}{P(I(x,y)=c)} P(\mathrm{ROI}) \quad (7)$$

where

$$P(I(x,y)=c) = P(I(x,y)=c \mid \mathrm{ROI})P(\mathrm{ROI}) + P(I(x,y)=c \mid \mathrm{BG})P(\mathrm{BG}) \quad (8)$$

where P(ROI) and P(BG) are constants, and the sum of them should be one. Bradski [15] recommends a value of 0.5 for them.

4. Simulation Results

A range of algorithms of the proposed hand mouse system are proposed. This section shows the simulation results. The algorithms can be classified according to their purpose. The algorithms in observe mode detect the target object, and the other algorithms track the target object.

Fig. 6 shows the result of each algorithm in observe mode. Fig. 6(a) and (b) shows the original image and the result of the frame voting algorithm, respectively. The man is shaking his hand, and his arm is detected as a moving object. On the other hand, by frame voting, some noise caused by vibrations is observed. Fig. 6(c) shows the result of the erosion-dilation algorithm of eliminating noise, so almost all of vibrations are removed. Fig. 6(d) shows the result of the sub-pixel-based DBSCAN algorithm and wrist elimination algorithm. Fig. 7 presents the results of the ABCshift algorithm for tracking the target object (a hand).

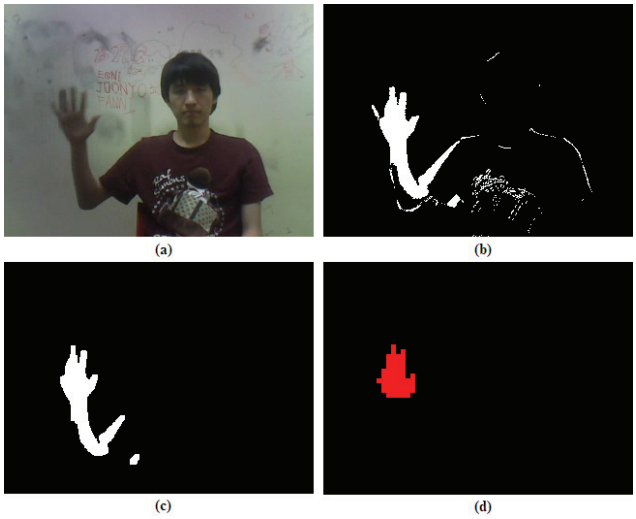

Fig. 6. The original image and results of the moving object detection when n=16 (a) The original image, (b) The result of frame voting algorithm, (c) The result of erosion-dilation algorithm, (d) The result of DBSCAN algorithm and wrist elimination algorithm.

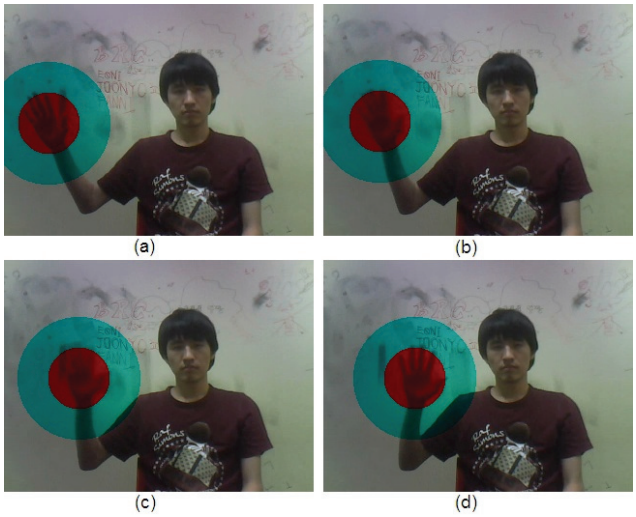

Fig. 7. The results of ABCshift algorithm. (a) n=86, (b) n=89, (c) n=92, (d) n=95.

5. Conclusion

This paper proposed a hand mouse system with little information on the situation. When information on the user and background is unavailable, the proposed system can succeed using only pre-defined gesture. Furthermore, the system has low computational complexity. On the other hand, the proposed system still has some issues, such as collisions between moving objects, which need to be addressed before it can be used in practical situations.

References

[1] N. Soontranon, S. Aramvith, and T.H. Chalidabhoingse, "Improved Face and Hand Tracking for Sign Language Recognition," *Proceedings of the Information Technology: Coding and Computing*, Vol.2, pp.141-146, 2005. Article (CrossRef Link)

[2] V. Athitsos and S. Sclaroff, "An appearance-based framework for 3D hand shape classification and camera viewpoint estimation," *Proceedings of the Automatic Face and Gesture Recognition*, pp.45-50, 2001. Article (CrossRef Link)

[3] R. Bunelli, "Template matching techniques in computer vision: theory and practice," Wiley, ISBN 978-0-470-51706-2, 2009. Article (CrossRef Link)

[4] A. K. Jain, B. Klare, and U. Park, "Face Recognition: Some Challenges in Forensics," *Proceedings of the Automatic Face and Gesture Recognition and Workshops*, pp.726-733, 2011. Article (CrossRef Link)

[5] C. Oh, M. Z. Islam, J. Park, and C. Lee, "A gesture recognition interface with upper body model-based pose tracking," *Proceedings of the Computer Engineering and Technology*, Vol.7, pp.V7-531-V7-534, 2010. Article (CrossRef Link)

[6] S. Kim, S. Kim, Y. Yoo, S. Lee, M. Ha and J. Yi, "Real-time motion recognition algorithm based on frame differences", *Proceedings of Institute of Electronics Engineers of Korea*, Fall conference, Vol. 32, No. 2, pp.215~216, 2009. Article (CrossRef Link)

[7] M. Oral and U. Deniz, "Centre of mass model – A novel approach to background modeling for segmentation of moving objects," *Proceedings of Image and Vision Computing*, Vol. 25, pp. 1365-1375, 2007. Article (CrossRef Link)

[8] E. R. Dougherty, "An introduction to morphological image processing," SPIE – International Society for Optical Engine, ISBN 0-8194-0845-X, 1992.

[9] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.24, No.7, pp.881-892, 2002. Article (CrossRef Link)

[10] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Department of Electrical Engineering and Computer Science, U.C. Berkeley, Berkeley, CA 94704, TR-97-021, 1998. Article (CrossRef Link)

[11] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining FLEXChip Signal Processor*, pp.226-231, 1996. Article (CrossRef Link)

[12] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257-286, 1990. Article (CrossRef Link)

[13] G. J. Allen, Y. D. Richard and S. J. Jesse, "Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces," *Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing (VIP 2004)*, Vol. 36, pp. 3-7, 2004. Article (CrossRef Link)

[14] R. Stolkin, I. Florescu, M. Baron, C. Harrier and B. Kocherov, "Efficient visual serving with the ABCshift tracking algorithm," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2008)*, pp. 3219-3224, 2008. Article (CrossRef Link)

[15] G. R. Bradski, "Computer video face tracking for use in a perceptual user interface," *Intel Technology Journal*, Q2 1998. Article (CrossRef Link)

Kyung-Won Kim received his B.S. degree in electronics engineering from Korea University, Seoul, Korea, in Feb. 2009. He is currently pursuing his Ph.D. degree at the department of computer and radio communications engineering at the same university. His research interests include computer vision and pattern recognition.

Daehee Bae received his B.S. degree in computer engineering from Kwangwoon University, Seoul, Korea, in Feb. 2011. He is currently pursuing his M.S. degree at the department of computer engineering at the same university. His research interests include SoC and system-level power analysis and optimization.

Joonhwan Yi received the B.S. degree in electronics engineering from Yonsei University, Seoul, Korea, in 1991, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 1998 and 2002, respectively. From 1991 to 1995, he worked with Semiconductor Business, Samsung Electronics Company, Korea, where he was involved in developing application specific integrated circuit cell libraries. From 2003 to 2008, he was with the Telecommunication Network, Samsung Electronics Company, Korea, where he worked on system-on-a-chip designs. Since 2008, he has been a faculty member of the Computer Engineering Department at Kwangwoon University, Seoul, Korea. His current research interests include C-level system modeling for fast hardware and software co-design, system-level power analysis and optimization, behavioral synthesis, and high-level testing.

Seong-Jun Oh is an Associate Professor at the Department of Computer and Communications Engineering, Korea University, Seoul, Korea. Before joining Korea University in September 2007, he was with Ericsson Wireless Communication, San Diego, California, USA as a senior Engineer from September 2000 to March 2003. He was also with Qualcomm CDMA Technologies (QCT), San Diego, California, USA as a Staff Engineer from September 2003 to August 2007. He received his B.S. (magna cum laude) and M.S. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1991 and 1995, respectively, and received his Ph.D. at the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor in September 2000. He served in the Korean Army during 1993-1994.

His current research interests are in the area of wireless/mobile networks with emphasis on the resource allocation for next-generation cellular networks with the physical-layer modem implementation. While he was with Ericsson Wireless Communication, he was an Ericsson representative for WG3 (physical layer) of the 3GPP2 standard meeting. While in QCT, he developed CDMA modems in ASIC for the base station (CSM 6700) and mobile station (Qualcomm Interference Cancellation and Equalization, QICE). From 2008 to 2010, he served as a Vice-Chair of TTA PG 707, the Korean evaluation group registered in ITU-R, where he is in charge of performance evaluations of LTE-Advanced and IEEE 802.16m systems, submitted as an IMT-Advanced technology in ITU-R WP-5D. He has received the Seoktop Teaching Awards from the College of Information and Communication, Korea University for outstanding lectures in the fall semester of 2007 and spring semester of 2010. He was a recipient of the Korea Foundation for Advanced Studies (KFAS) Scholarship from 1997 to 2000.