The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Fingertip Detection with Morphology and Geometric Calculation

Dung Duc Nguyen, Thien Cong Pham and Jae Wook Jeon, *Member, IEEE*
ICE School, Sungkyunkwan University, Suwon, Korea 440-746
nddunga3@skku.edu, pham@ece.skku.ac.kr, jwjeon@yurim.skku.ac.kr

***Abstract*— We present a method to detect human fingertips from images captured by a stereo camera. The system makes use of the disparity information from a stereo camera to find candidates, and defines an evaluation process to detect two hands. The finger detector then processes each hand image to extract finger images. Finally, we perform geometric calculations on the results to relocate the positions of the fingertips. The proposed method is not complex; however, it shows exciting results in terms of run time and detection rates. The extraction result can be used in hand configuration modeling for gesture recognition in HCI systems.**

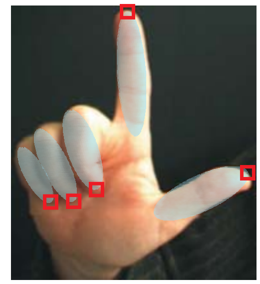


Fig. 1. The hand and expected position of fingertips.

I. INTRODUCTION

For years, people have used traditional devices such as a keyboard, mouse, pointing device, etc. to interact with computer systems. These communication methods are unnatural to humans. Therefore, people tend to find other ways to communicate with computer systems. For example, one can use his voice to control a computer, or he can move his hands in front of a camera to give some commands. Constructing such intelligent systems presents challenges in human-computer interaction (HCI) systems. There are many applications using voice recognition techniques to improve communication between humans and the system. However, gesture recognition techniques must deal with many problems related to detection and recognition, and work on such systems is ongoing.

Most of gesture recognition systems use trajectories of hand motion and hand configuration. Identifying a hand gesture can be done in many ways, depending on the problem to be solved. One way is to use a learning scheme and build a classifier such as SVM, as in [3], [14]. These kinds of statistical methods are fundamental for this field. In addition to classifiers, hand structure recognition methods are also powerful tools, even if they are difficult to deal with. The key issue in these techniques is the definition of the hand structure model. In some complex systems such as GREFIT [9], [10], the hand model is defined by links and joints. This approach uses fingertips, which are usually detected by applying a learning model, to reconstruct the hand model. The detection of human fingertips is an important issue in most hand model studies and in some gesture recognition systems.

This paper describes a fast method to detect fingertips. We overcome problems of other studies based on hand boundaries [1], [7], [8], [5], [2] and image quality. The proposed system takes advantage of depth information provided by a stereo camera and a modified skin detector to extract hand regions. We apply some constraints to the hand, and process each hand individually to detect finger-like areas. Our process makes use of grayscale morphology and geometric calculations to relocate fingertip locations. A demonstration of this idea is shown in Fig. 1.

This work is partially supported by Samsung Electronics

II. RELATED WORK

Most finger detection studies require hand regions as a starting point. In [2], [5], [6], [11], [14], authors extract a hand region from the scene using segmentation techniques. In a color image, using skin color is more appropriate than hand shape since the shape can be recovered from the skin region. In [13], Vezhnevets presents popular methods for skin modeling and detection. From comparisons in [13], the mixture of Gaussian model produces good results among other skin detection methods. In this work, we use a three-dimentional (3-D) Gaussian model and modify the way of choosing the threshold to achieve a better skin detection result.

In [11], Schmugge uses a classifier incorporating Bayesian decision theory for skin detection. He then processes skin regions using the hand detector provided by Shin *et al.* [12] to take out the hand from the remainder. The authors remove noise and rejects regions that are small or have unlikely texture or shape. The hand is detected by choosing the closest region from the range image (i.e. the disparity image). We use a similar approach to detect two hands from the scene by making use of the range image.

JongShill [7] detects the difference between two consecutive frames and measures the entropy to extract the hand from the image instead of using a single image. He uses contour to classify hand postures. Other learning-based

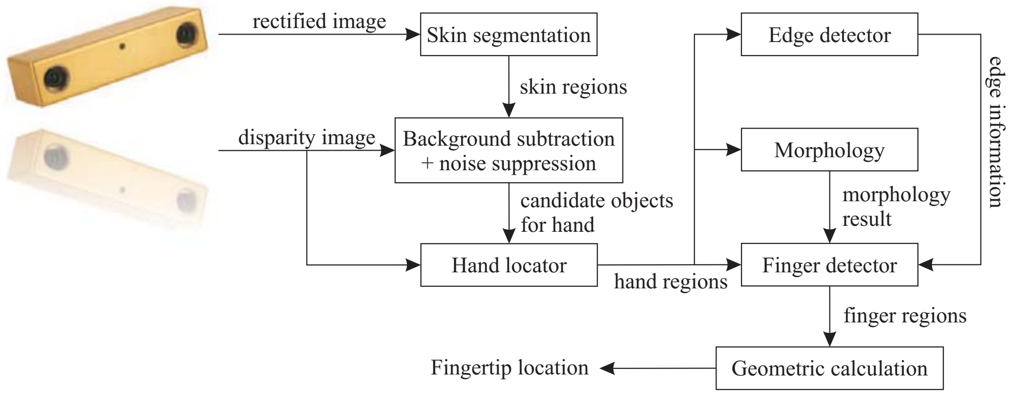


Fig. 2. Block diagram of the proposed method.

recognition methods are introduced in [3], [14]. Another way to detect hand posture is analyzing the hand structure. If there are good detectors which can detect hand features such as fingers, then hand posture can be better represented by hand structure models. Studies modeling the human hand for gesture recognition are cited in [9], [10]. These studies try to reconstruct the hand model from detected fingers using the inverted kinetic model. The human finger is the key point in hand posture recognition using hand structure modelling.

A common method for finger detection uses the Gabor feature vector. In [9], [10], the authors use Gabor filters to extract features and detect fingertips by using a local linear mapping (LLM) network. They then compute parameters of hand structure using parameterized self-organizing map (PSOM) to reconstruct the 3D hand shape. Kerdvibulvech [6] uses a similar approach to detect the fingertips of a guitar player. He uses six Gabor filter kernels and an isotropic Gaussian to capture the feature vector, and then makes use of LLM for both global and local processing.

Other finger detection techniques that do not use feature vector were introduced in [2], [5], [7], [8]. In [7], after extracting the hand region from the image sequence, the author uses (r, θ) plot to detect fingers from the hand boundary. This method is commonly used in many finger detection methods. However, it can only detect some opened fingers, and recognizes some hand postures since it discards some hand information such as texture inside the hand. A similar idea about hand recognition based on hand shape appeared in [8]. The authors detect five fingertips and four inter-finger points from hand curvature using B-spline fitting. Jiang [5] also uses an approach similar to [7]; that is, he takes the gradient gram of the contour for initial fingertip positions and refines each fingertip location using radial distance maxima. These types of methods can only detect fingers appearing at the boundary. In [2], the authors use curvature information to detect fingertips and refine the result by considering the correspondence between arcs and fingers. This step, and the estimation of undetected arc points, makes the algorithm more robust.

Barrho [1] approaches the fingertip detection problem from another viewpoint. He makes use of generalized Hough transform (GHT) to match a fingertip template with a hand image. The result still has noise (false detection) since there are many regions which have the same shape as the fingertip. So far, we have described several studies on finger detection and have shown limitations of these methods. In the next section, we detail our approach to fingertip extraction and discuss experiments to show the efficiency of our method.

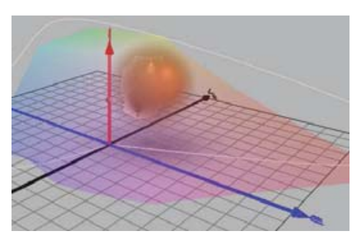
III. THE DETECTION PROCESS

The proposed method is described briefly in the Fig. 2. We use a stereo camera to capture images. First, the image is processed by skin detection module to extract skin regions. The second step is background subtraction and noise suppression. We obtain disparity image from the camera and combine with detected skin regions to find candidates for hand-liked objects. The hand locator module uses an evaluation process to find two hands. Then, the finger detector takes the edge image and morphology result as inputs for finding finger regions. We perform geometric calculations in the last step to relocate finger locations; in another word, we try to find the position of fingertips from these finger objects.

A. Skin Segmentation

People usually recognize a hand by its color and shape. Therefore, we chose the skin segmentation technique to ex-

(a)



(b)

Fig. 3. (a) Skin samples from our images, (b) The skin color distribution of training data.

tract skin regions. This technique is widely used in most hand recognition methods. The comparisons in [13] show that the mixture-of-Gaussian model is a good approximation model for human skin. The skin color distribution is represented by the following density function

$$p(\mathbf{c}|skin) = \sum_{i=1}^{k} \pi_i p_i(\mathbf{c}|skin), \quad (1)$$

where k is the number of components and π_i are the weight factors of each component. We use the CIELUV color space to represent skin color instead of RGB color space. We use a function to measure color similarity based on Mahalanobis distance:

$$d_i(\mathbf{x}) = (\mathbf{x}-\mu_i)^T \Sigma_i^{-1} (\mathbf{x}-\mu_i) + \ln|\Sigma_i| - 2\ln\pi_i. \quad (2)$$

A pixel is classified as skin if above expression is less than a specified threshold T_0 which we obtain by experiment. In above expression, we take both weight factor and determinant of covariance into account. It makes sense because if the weight factor π_i of a cluster is small, we should choose the color near the mean of that cluster as skin color. Similarly, we classify the color as skin color if it close to the mean of the cluster which has large variance, i.e. $\det\Sigma_i$ is large.

We collect skin samples by separating skin regions from captured images. The skin samples and the distribution of skin color are shown in Fig. 3. There are three main clusters in the color distribution. Therefore, it is sufficient to choose $k = 3$ to train the skin color. In our case, the threshold $T_0 = 55$ produces good results. The values of k and T_0 are not unique because they depend on the color distribution of the skin samples.

B. Hand detection

In some finger detection studies such as [9], [10], it is assumed that the hand has already been detected and thus a hand detection step is not needed. This assumption does not hold in the general case since the hand can be confused with other objects having the same color. Furthermore, skin detection result is not always perfect, which can cause a hand detection method based on shape to fail. Therefore, we propose two methods to detect human hand from above skin regions by taking the advantage of disparity information from the stereo camera.

In our experiments, a person acts in front of a stereo camera and moves his hand around. Two hands have relative large area and the depth of hand is in a specified range. This constraint depends on how good the disparity image is. If the disparity image is good, as shown in Fig. 4(a), then we can use it to reject background without degrading the quality of skin regions. In addition, we do not need to worry about the skin overlapping problem because we have already rejected occluded regions such as the face. However, the graph-cuts method consumes much time and thus is not suitable for real-time applications. For this reason, we use the disparity image obtained from Bumblebee camera. The quality of disparity image is not as good as in Fig. 4(a) but it is sufficient to approximate the depth of skin regions.

(a) (b)

Fig. 4. (a) Disparity image generated using the graph-cut method, (b) The hand extraction result.

We pre-process the skin regions using basic morphology operators to reduce small noises. Let $I_s(\mathbf{x})$, $I_b(\mathbf{x})$ be the skin image and the binary image resulting from skin image. The skin objects after noise suppression are represented by

$$I_{bs}(\mathbf{x}) = (I_b(\mathbf{x}) \ominus W_1(\mathbf{x})) \oplus W_2(\mathbf{x}), \quad (3)$$

where $W_1(\mathbf{x})$, $W_2(\mathbf{x})$ are kernel windows for the erode and dilate operations. This step reduces noises in $I_s(\mathbf{x})$ and significantly speeds up the object detection process.

We estimate the disparity of the object by taking the average over the region:

$$d_i = \frac{1}{N_i} \sum_{\mathbf{x}\in O_i} I_d(\mathbf{x}), \quad (4)$$

where $I_d(\mathbf{x})$ is the disparity image and d_i is estimated disparity of the object O_i, which has area N_i. As previously stated, the hand moves in a certain range; therefore, we reject any object whose disparity value d_i is not in the range. In this case, we inspect the range directly from the disparity image, and choose the range $[55;155]$ according to our test images.

In the first detection method, we propose an evaluation function which has the form

$$f_{hh}(A,B) = \sum_{i=1}^{k} w_i f_i(A,B), \quad (5)$$

where k is the number of criteria, and w_i is the weight factor associated with the function $f_i(A,B)$. We use three criterion functions to produce scores for two objects A and B: $f_{sc}(A,B)$ measures the similarity between two objects, $f_{sdc}(A,B)$ measures the difference in size, and $f_{dc}(A,B)$ measures the disparity criterion. These weight factors can be chosen by training through the supervising learning process. In simple case, we can set all weight factors w_i to one. Then, two objects giving the highest score to f_{hh} are considered to be hands.

We propose a second method to detect hands by applying rejection steps. In our experiments, two methods produce the same detection result in most cases. However, the second method is simpler. It performs the following steps:

- Arrange objects in descending order of area size.
- Reject small and large objects

$$S_1 = \{O_i | f_s(O_i) \in [a;b]\}.$$

(a) (b)

Fig. 5. Result of finger extraction using binary morphology. (a) Binary image of two hands, (b) Result subtracting (a) from the morphology image of (a).

- Reject objects which are not in the range given by

$$S_2 = \{O_i | O_i \in S_1 \wedge f_d(O_i) \in [d_l; d_h]\}.$$

- Filter candidates: we choose the three largest objects in range $[a;b]$. They are the head and two hands.
- Reject the human head: the head can be eliminated by checking the relative position and size compared to other objects. The two remaining objects are hands.

The hand extraction result is shown in Fig. 4(b). Although the first detector is heuristically based, it can be modified by applying other criterion functions and tracking techniques to form a better detector.

C. Finger detection

After extracting two hands, we perform a finger detecting process on each hand. As shown in Fig. 4(b), there are noises on the boundary and small holes inside the hands due to shadows. These noises have a significant affect on finger detection techniques based on the hand boundary, as cited in [2], [5], [7], [8]. Therefore, we propose a detection method which can deal with noise in a hand image.

The fingers are long and thin compared to the hand. Therefore, they disappear when we apply erode operator on the hand image. The open operator can be used to detect fingers by choosing an appropriate window and detecting the difference. We discard the color information of the hand and perform open operator on the hand image. The subtraction result produces the finger-like regions as in Fig. 5. It is apparent that this result is inadequate. The fingers are hard to detect due to noise, including the size and position of the noise. In Fig. 5(b), two fingers are stuck together and produce the wrong detection result.

We propose using grayscale morphology operator to overcome the problem. Because the hand does not have much texture, it is reasonable to convert hand image from skin color to grayscale. Since we use grayscale morphology, the hand texture is reduced but not discarded. Even if we reduce hand texture, the texture still affects the detection process. Let $I(\mathbf{x})$ be the grayscale patch of the hand. The morphology image can be defined formally as

$$I_{gmO}(\mathbf{x}, d, \sigma) = f_{gmO}(g(\sigma) * I(x), W(d)), \quad (6)$$

where $f_{gmO}(.)$ represents the grayscale open operator, $g(\sigma)$ is the Gaussian kernel with standard deviation σ, and $W(d)$ is the morphology window defined by the disparity value d of the hand.

We perform many experiments and find out that the Gaussian kernel has an important role in our detector. It reduces the noise on the boundary and diffuses the color information of the hand. The effect of diffusion makes finger response smooth and stable even if we do not have good hand image. In our tests, the detection is superior when we apply Gaussian filter. We choose $\sigma = 1$ for the Gaussian function and kernel size 9 for the open operator. When the hand moves back and forth, the hand size changes by a scale factor. Therefore, we take disparity information into account by using $W(d)$ instead of a fixed window size.

Let $E(\mathbf{x})$ be the edge image detected from the hand patch, and $Th(.)$ be a threshold function. The finger response R_f is calculated as follows:

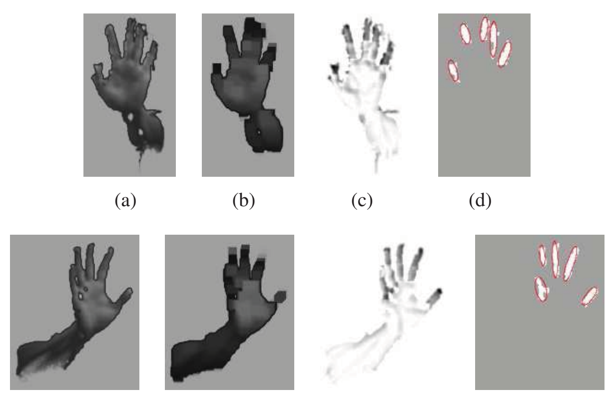
$$\begin{aligned} S(\mathbf{x}, d, \sigma) &= g(\sigma) * I(\mathbf{x}) - I_{gmO}(\mathbf{x}, d, \sigma), \\ R_f(\mathbf{x}, d, \sigma, th) &= Th(S(\mathbf{x}, d, \sigma) \wedge \neg E(\mathbf{x}), th), \end{aligned} \quad (7)$$

where $S(.)$ is the subtraction result and th is the threshold for finger response. We process the $S(.)$ image using a threshold filter to reduce noise and to separate the fingers from each other. In most of our experiments, the threshold value th varies at around 75% of the maximum value of $S(.)$. Therefore, we choose 0.75 for threshold th and apply object analysis on the resulting image patch.

Fig. 6 shows the finger extraction result. From the response image, we estimate the position, direction, and length of the fingers. Combining with the mass center, we relocate the position of the fingertips and hand center.

D. Relocating finger position

First, we relocate the hand center. Assume that the ratio between the major axis and minor axis of the hand palm is β. We use vectors $\mathbf{v}_0$ and $\mathbf{v}_1$ to find the moving angle. Vector $\mathbf{v}_0$ is the direction vector of the major axis, and points to the



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

(e) (f) (g) (h)

Fig. 6. Result of finger extraction using grayscale morphology operators and object analysis. Each row presents the result of each hand. (a), (e) The hand is taken out from the image, converted to grayscale, and smoothed with Gaussian kernel $\sigma = 1$. (b), (f) The result of grayscale open operator with kernel size 9. (c), (g) Subtracting result. (d), (h) Detected fingers.

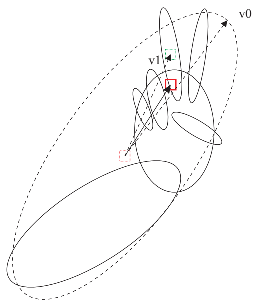


Fig. 7. The hand center after relocation (the solid red square).

estimated average point of fingers $\mathbf{x}_{eaf}$ which is given as the weighted sum

$$\mathbf{x}_{eaf} = (\sum_{i=1}^{N_f} \mathbf{x}_i s_i)/(\sum_{i=1}^{N_f} s_i), \tag{8}$$

where s_i are size of finger regions and N_f is number of detected fingers. The vector $\mathbf{v}_1 = \mathbf{x}_{eaf} - \mathbf{x}_0$ points to $\mathbf{x}_{eaf}$. Let θ_0 and θ_1 be the angles between $\mathbf{v}_0$, $\mathbf{v}_1$ and the horizontal axis. By applying a weight factor α, we can estimate the center of hand palm as follows

$$\theta = \alpha\theta_0 + (1-\alpha)\theta_1, \tag{9}$$

$$\mathbf{x}_e = \mathbf{x} + (l_{ma} - \beta l_{mi})[\sin\theta \quad \cos\theta]^t, \tag{10}$$

where $\mathbf{x}$ is the previous position of the hand. We adjust the hand angle by the direction vector $\mathbf{v}_h = \mathbf{x}_{eaf} - \mathbf{x}_e$. Using the new center, we reject any finger region out of range; i.e., the ith finger should be rejected if $\mathbf{v}_{ic}^t \mathbf{v}_h < 0$, where $\mathbf{v}_{ic}$ is the vector from the hand center $\mathbf{x}_e$ to the ith finger center.

We use the distance gap between open fingers and closed fingers to classify them. Let r_i be the distance from the finger center to hand center $\mathbf{x}_e$. We consider a finger to be a closed finger if r_i is less than the threshold r_{th} chosen by inspecting the distance from the fingers to $\mathbf{x}_e$. In the more general case, we can use $r_{th}(d)$ as a function of depth.

Let $\mathbf{x}_i$, θ_i be the center and the angle of the ith finger. Vectors $\mathbf{v}_i$ and $\mathbf{u}_i$ are defined as follows:

$$\mathbf{v}_i = \mathbf{x}_i - \mathbf{x}_e,$$
$$\mathbf{u}_i = [\sin\theta_i \quad \cos\theta_i]^t.$$

We classify the status of a finger as unknown if the value $\mathbf{v}_i^t \mathbf{u}_i$ is less than $c_0|\mathbf{v}_i|$. We choose 0.5 as the value of the constant c_0. The fingertip location in this case is the same as the finger center.

For an open finger, the value $|\mathbf{v}_i|$ is greater than the threshold $r_f(d)$, and the finger length l_i is not below $l_f(d)$. In the normal case, when two hands are placed as shown in Fig. 4(a), we select $r_f = 40$ and $l_f = 30$. The location of the fingertip is estimated as follows:

$$\mathbf{x}_{ef,i} = \mathbf{x}_i + \mathrm{sign}(\mathbf{v}_i^t \mathbf{u}_i) 0.5 l_i \mathbf{u}_i. \tag{11}$$

Similarly, the estimation of closed fingers is given as follows:

$$\mathbf{x}_{ef,i} = \mathbf{x}_i - \mathrm{sign}(\mathbf{v}_i^t \mathbf{u}_i) 0.5 l_i \mathbf{u}_i. \tag{12}$$

	Left hand (ms)	Right hand (ms)
Load image	07.83	
Pre-process	20.92	
Detection	12.75	13.08
Total	54.58	

TABLE I

THE PROCESSING TIME OF OUR ALGORITHM.

IV. EXPERIMENTAL RESULTS

We carry out experiments using a single PC and a stereo camera. The program uses one thread and runs on CPU AMD Althlon 4800+. We use the frame resolution of 640×480 provided by the camera. Our camera does not give high quality images; therefore, noise tolerance is an importance requirement.

For the experiment shown in Fig. 6, the hand image has a lot of noise on hand boundary, some holes inside, and two fingers are stuck together. Therefore, the methods described in [2], [5], [7], [8] fail to detect fingers because they are highly depend on the hand boundary. In addition, the quality of skin image is affected by light condition, and may cause noise in hand image. Our method shows good detection results in these experiments.

The run-time of the method is shown in Table. IV. Preprocessing steps include skin extraction, grayscale conversion, threshold checking and binary morphology. The detection steps including hand extraction, finger extraction, and fingertip location do not consume much time.

To measure recognition rate, we observe 120 frames to check the performance of our detector. For open fingers, the detection rate is approximate 90-95% accuracy while closed fingers can be detected with a lower rate (about 10-20%). According to our results, there are two main reasons cause this low detection rate: image quality and morphology operator. Since we can not change the image quality of camera, we need to find a suitable response function. The function should have ability to cover the texture inside morphology window because the hand does not have much texture.

In the second experiment, we use some images from Massey's database [4] to test the method. The Fig. 8 shows the finger extraction result with two closed fingers detected and small response of the thumb. In this experiment, methods

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

Fig. 8. The result of finger extraction. (a) Image from Massey databased [4] converted to grayscale, (b) Result of grayscale open operator, (c) Subtraction result, (d) Finger extraction result.

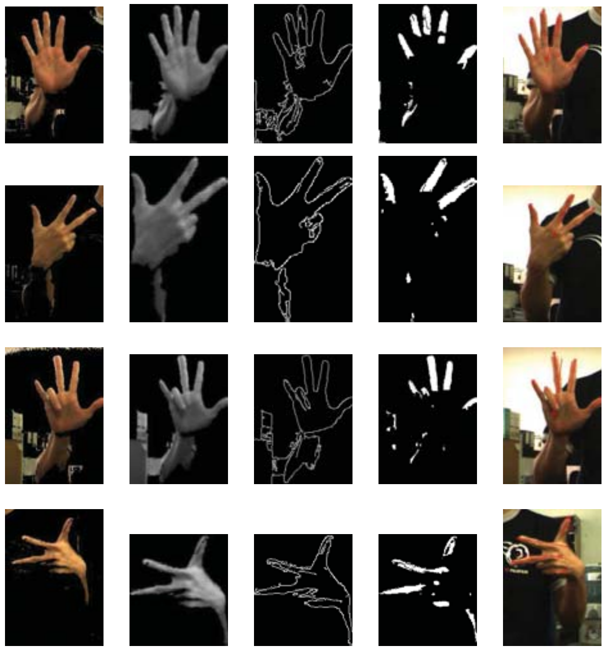


Fig. 9. Detection results. Fingertip locations are small squares and the hand's center is the big square. The lines from the center represent hand and finger directions. We cropped the images from the original result. Each line is the detection result on the hand. From left to right: skin segmentation, smoothed grayscale image, edge image, finger response image, and final location and direction of fingertips.

in [2], [5], [7], [8] cannot detect two closed fingers. The hand boundary is good, however, using the previously described detection techniques, which are based on hand boundaries, will fail because the closed fingers do not appear on the boundary. In such cases, the (r, θ) plot can only help detect open fingers. The B-spline fitting method in [5] also fails for closed fingers. Some improvements in [2] may help to detect closed fingers. However it uses arcs on the contour, and it may fail with noise images such as in Fig. 6.

Fig. 9 shows additional experimental results with various hand postures. The images are taken in evening when light condition makes captured color of many objects in the room look like skin color. In addition, the skin color of the hand is not perfect, as we can see in the first column (there is considerable noise on the hand). It is clear that we cannot get the full shape of the hand for exact calculation. In these cases, methods based on hand shape will fail. Our method, however, may detect fingertips in some cases. The detection results depend on our finger response function. Therefore, it is possible to improve the detection method by finding a function that combines the gradient and texture information inside the hand.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a simple and effective method to detect human fingertips. The method can deal with noise in skin images and solve the problems of previous methods that are based on curvature measurement and hand shape. By performing simple operations, our method shows good performance in terms of run-time, and is suitable for real-time applications. Detected fingertips can be used for hand configuration modeling and gesture recognition, especially in HCI and robot applications.

As noted in Section IV, our method has some limitations in detecting closed fingers. Therefore, we need some improvements in the method. Future work will include the modification of the local window function, which can use both gradient and texture information inside the hand. In addition, an adaptive skin detector will be applied to produce a better skin detection result.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support from Samsung Electronics during this project.

REFERENCES

[1] J. Barrho, M. Adam, and U. Kiencke. Finger localization and classification in images based on generalized hough transform and probabilistic models. In *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–6, 2006.

[2] W. Chen, R. Fujiki, D. Arita, and R. ichiro Taniguchi. Real-time 3d hand shape estimation based on image feature analysis and inverse kinematics. In *Proc. of the 14th International Conference on Image Analysis and Processing (ICIAP)*, pages 247–252, Washington, DC, USA, 2007. IEEE Computer Society.

[3] Y.-T. Chen and K.-T. Tsengn. Developing a multiple-angle hand gesture recognition system for human machine interactions. *33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 489–492, 2007.

[4] F. Dadgostar, A. L. C. Barczak, and A. Sarrafzadeh. A color hand gesture database for evaluating and improving algorithms on hand gesture and posture recognition. In *Research Letters in the Information and Mathematical Sciences*, volume 7, pages 127–134, 2005.

[5] X. Jiang, W. Xu, L. Sweeney, Y. Li, R. Gross, and D. Yurovsky. New directions in contact free hand recognition. In *International Conference on Image Processing (ICIP)*, volume 2, pages 389–392, 2007.

[6] C. Kerdvibulvech and H. Saito. Vision-based detection of guitar players' fingertips without markers. In *Proc. of the Computer Graphics, Imaging and Visualisation (CGIV)*, pages 419–428. IEEE Computer Society, 2007.

[7] J. Lee, Y. Lee, E. Lee, and S. Hong. Hand region extraction and gesture recognition from video stream with complex background through entropy analysis. *Proc. of the 26th Annual International Conference of the IEEE EMBS*, 2004.

[8] Y. Ma, F. Pollick, and W. T. Hewitt. Using b-spline curves for hand recognition. In *Proc. of 17th International Conference on the Pattern Recognition (ICPR)*, volume 3, pages 274–277, Washington, DC, USA, 2004. IEEE Computer Society.

[9] C. Nölker and H. Ritter. Parametrized soms for hand posture reconstruction. In *International Joint Conference on Neural Networks (IJCNN)*, volume 4, pages 139–144, 2000.

[10] C. Nölker and H. Ritter. Visual recognition of continuous hand postures. *IEEE Trans. Neural Networks*, 13(4):983–994, 2002.

[11] S. Schmugge, M. A. Zaffar, L. V. Tsap, and M. C. Shin. Task-based evaluation of skin detection for communication and perceptual interfaces. *Journal of Visual Communication and Image Representation (JVCIR)*, 18(6):487–495, 2007.

[12] M. C. Shin, L. V. Tsap, and D. B. Goldgof. Gesture recognition using bezier curves for visualization navigation from registered 3-d data. *Pattern Recognition*, 37(5):1011–1024, 2004.

[13] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *Proc. of 13th International Conference of Computer Graphics and Visualization Graphicon*, pages 85–92, 2003.

[14] Y. Wu and T. S. Huang. View-independent recognition of hand postures. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 88–94, 2000.