# QPalm: A Gesture Recognition System for Remote Control with List Menu

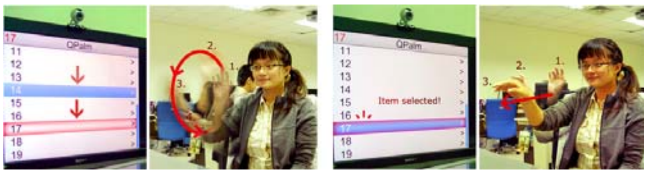
Yu-Hsin Chang, Li-Wei Chan, Ju-Chun Ko, Ming-Sui Lee, Jane Hsu, Yi-Ping Hung
Graduate Institute of Networking and Multimedia, National Taiwan University
email: hung@csie.ntu.edu.tw

***Abstract*—The coming ubiquity of digital media content is driving the need of a solution for improving the interaction between the people and media. In this work, we proposed a novel interaction technique, QPalm, which allows the user to control the media via a list menu shown on a distant display by drawing circles in the air with one hand. To manipulate a list menu remotely, QPalm includes two basic functions, browse and choosing, realized by recognizing the user's palm performing circular and push motions in the air. The circular motion provides fluidity in scrolling a menu up and down, while push motion is intuitive when the user decided to choose an item during a circular motion. Based on this design, we develop a vision system based on a stereo camera to track the user's palm without interfering by intruders behind or next to the operating user. For more specifically, the contribution of the work includes: (1) an intuitive interaction technique, QPalm, for remote control with list menu, and (2) a palm tracking algorithm to support QPalm based on merely depth and motion information of images for a practical consideration.**

***Index Terms*—remote control, selection technique, human-computer interaction, gesture recognition**

## I. INTRODUCTION

Interaction design generally refers to the discipline of defining the behavior of products and systems that a user can interact with. Many interaction techniques are developed to help the user manage jobs more intuitive, effective, and natural. Some methods use a substantial device, like a pen, to control the media content [4][2], while others simply use a gesture to make a command [1] [6]. The EyeToy, devised in 2003, is developed for playing the games on PlayStation 2 by using a camera sensor. The Eyetoy allows the players to interact with games by their hand motions and sound without the need of holding a game controller. Wii, another popular game console was released by Nitendo in September, 2006. A distinguishing feature of the console is its wireless controller, the Wii mote, which can be used as a hand-held pointing device in which the built-in sensor can detect its motions in three dimensions. In addition to game industry, digit home situates another scenario requiring sophisticated interaction design. From air condition to television, most home appliances can be controlled via a control panel built-in or a remote controller. Not only would the user be confused by multiple controllers, but these appliances might share different metaphors in-between designs could also confuse its users. It is apparent that developing new solutions is demanding for next generations.



In this work we propose a new type of interaction technique, QPalm, for remote control with list menu. The interaction technique is based on recognizing the users' hand motion by using a stereo camera so that the user has no need to take any controller. In the proposed design, all information is gathered into a list menu, while the menu is displayed on a screen distant to the user. QPalm allows the users to browse and to choose an interested item in the list menu by simply drawing circles in the air with their hands. Specifically the browse and the choosing functions for the interaction are described as follows. Browse function allows the users to control the list menu by

Manuscript received October 9, 2001. (Write the date on which you submitted your paper for review.) This work was supported in part by the U.S. Department of Commerce under Grant BS123456 (sponsor and financial support acknowledgment goes here). Paper titles should be written in uppercase and lowercase letters, not all uppercase. Avoid writing long formulas with subscripts in the title; short formulas that identify the elements are fine (e.g., "Nd–Fe–B"). Do not write "(Invited)" in the title. Full names of authors are preferred in the author field, but are not required. Put a space between authors' initials.

F. A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (corresponding author to provide phone: 303-555-5555; fax: 303-555-5555; e-mail: author@ boulder.nist.gov).

S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu).

T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan (e-mail: author@nrim.go.jp).

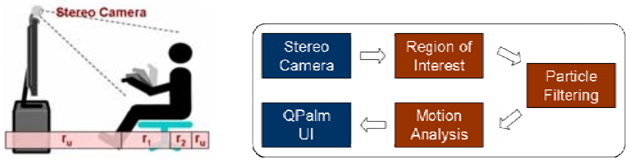


Fig. 2. (a) The system environment from a side view. (b) Our system overview.

making circular motions with one hand. Drawing a circle clockwise scrolls the highlighted item down and vice versa (Figure 1(a)). Choosing function is activated by making a push motion, once the user reaches the target item (Figure 1(b)). The combination of a circular motion and a push motion seems drawing the letter 'Q' in the air so that the proposed work is named as QPalm.

QPalm is devised primarily working with list menu because list menu is one of the most commonly-used user interfaces. Most information and functions can be easily organized into a hierarchical list menu, and therefore can be manipulated remotely via QPalm. To efficiently track the users' hands with robustness, we develop a hand motion detection algorithm based on motion and depth information of the images taken by a stereo camera. Without considering color information of the image makes our system practical to be applied in the real world as the system would not be harmed by the users' colorful wearing and would work in most environments with different lighting conditions. For more specifically, the contributions of the work include: (1) an intuitive interaction technique, QPalm, for remote control with list menu, and (2) a hand motion detection algorithm to support QPalm based on merely depth and motion information of images for a practical consideration.

## II.Related Work

Recent years there are some products or researches which use the circular motion as their main user interface. The most famous example is the iPod1, a portable media player which is designed and released by Apple in October, 2001. The most well-know feature of this product is the click wheel, which is used as a rotational manner to scroll through menu items and control the volume. There is a click wheel and an anti-aliasing graphics display placed on the device. A user simply puts the thumb on the click wheel and moves along the circular wheel and the highlighted item in the menu then moves up and down. In their design, to browse in a list menu is easier since a smooth motion is performed.

There are also some circular-like interfaces applied to practical applications like navigating document. The virtual scroll ring (VSR) [2], which is proposed by Tomer Moscovich et al., is a scrolling technique that uses an existing general positioning device such as a touchpad, stylus or standard mouse to navigate the document. Clockwise motions scroll the view down and counterclockwise motions scroll it up. Besides, with the VSR, fast and large movements produce fast scrolling, while small and slow movements yield slow scrolling. The VSR adopts the length traveled along the circumference of a circle to decide the distance of the view that should scroll. In the experiment, the VSR based on a touchpad and mouse is compared with the mouse wheel while scrolling in a document. The results show that the VSR is a tenable scrolling alternative, especially when most scrolling actions are expected to be longer than half a page. Another approach, which is similar to the VSR, is proposed by G.M. Smith et al. and called the Radial Scroll [3]. The Radial Scroll uses the same design of motions as VSR but different in the functions of controlling the speed of scrolling. If users wish to scroll more slowly, they draw a larger circle; to scroll more quickly, they draw a relatedly smaller circle instead. An improved version of Radial Scroll is purposed by the same author, called Curve Dial [4]. As shown in Fig.2.2(b), the Curve Dial use the angle formed by the last three points of the trajectory to judge the size of the circles, and make responses immediately. This technique offers the eye-free parameter entry that absent in the Radial Scroll. In the experiment, their work is more suitable for navigating short-distance targets while traditional methods work better for browsing long-distance targets.

## III. Palm Detection and Tracking

In this work, the circular and push motions are the basic gestures designed for QPalm. The system is devised keeping reacting while the user is drawing a circle, rather than notify the user after an entire circle is accomplished. In this section we introduce the approach to detect the position of user's palm in real-time, followed by the analysis of the palm motions is described in next section. The workflow of the system is shown in Fig.2(b). Firstly, the region of interest is defined in the image frame according to the position and depth of the user's face. Next, useful features are extracted for particle filtering step to further locate the user's palms. The trajectories of detected palms are then adopted for motion analysis. Once a pre-defined motion is found, the QPalm interface gives visual and sound feedbacks to the user.

The environment of our system is illustrated in Fig.2(a). It is like an ordinary scenario in a living room: a user sitting on the sofa, a television placing in front of the user. To detect the trajectories of the user's palms in three dimensions, a stereo camera released by Videre Design is

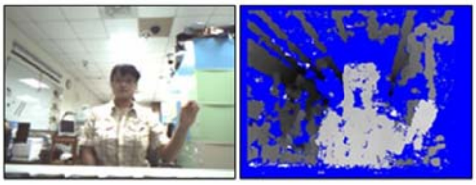


Fig. 3. The original color and stereo images at frame *t*.

introduced in the system, laying on the television. This camera provides disparity computation on hardware hence we can obtain the depth and color information of an object in the scene in real-time. One can also use a pair of commercial cameras to compute depth information, and one of them to provide color information. Fig.3 shows the color and depth images captured by the stereo camera. Noted that the blue region of the depth image indicated the depth is un-defined due to an un-matched condition occurred in disparity computation.

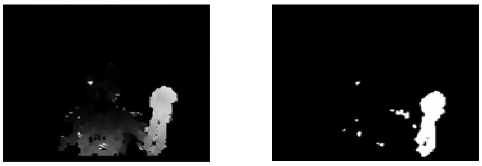
### *A. Preprocessing with Face Detection*

Firstly, we use Adaboost algorithm [8] to locate the user's face. Once a face is found, the system is triggered and Meanshift [9] algorithm is applied to track the face. Adaboost is generally more computation demanding to locate a face from an input image in comparison to Meanshift which simply searches a similar distribution locally. Combining of the two approaches leads to an efficient face tracking method to give a good region-of-interest for later computation. However, in case Meanshift could fail in a local search while tracking a face, Adaboost algorithm is applied to verify the tracked face positions every three seconds.

### *B. Identifying Region-of-Interest*

While an user is sitting and watching television, there are three parts in the side view, which are the region of the user's face $r_2$, the region of the user's hand while performing a motion, $r_1$, and the non-interested regions labeled $r_u$. A side view illustration is shown in Fig.2(a). Since $r_1$ always attaches to $r_2$, we have a general idea of the range of $r_1$ if $r_2$ is known. Here the range of $r_2$ is defined as the depth of the user's face, modeled as a Gaussian distribution.

The length of $r_1$ where users' hand may appear is defined as the arm length of a normal human in the initial. However, when the user performs a circular motion, the user's palm can easily drifts in depth during even the same sequence of the circular motion. To alleviate the drift effect, we use a mixture of Gaussian to model $r_1$. Since the range is drifting in a slow way, two or three Gaussians are sufficient to achieve this task. Additionally, we take the distance between $r_1$ and $r_2$ as the Gaussian mean instead of using a fixed value. Eq.1 shows the probability of a pixel belonging to $r_1$.



(a)Probability Map (b) Region-Of-Interest

Fig. 4. Use the depth information to obtain the probability map and the region-of-interest.

$$P(x \in r_1, t) = \sum_{i=1}^{K} \omega_{i,t} * \eta\left(d_{x,t}, \mu_{i,t}, \sigma_{i,t}\right) \qquad |(1)$$

, where $d_{x,t}$ is the distance of depth between *x* and mean of $r_2$. $K$ is the number of Gaussian distributions and $\omega_{i,t}$ is an estimate of the weight of the $i^{th}$ Gaussian in the mixture at time *t*. After applying Eq.1 to all pixels, we get a probability map tells the probability of a pixel belonging to user's hand, namely $P_t$. The image $B_t$, obtained by truncating pixels with low probability in $P_t$, is extracted to define the region of interest and give a rough shape of a hand (see Fig.4). This method takes the structure of human body into consideration, making the detection result more reliable. The final detecting result is used to update the mean of the mixture of Gaussian which represents $r_1$.

### *C. Palm Tracking Using Particle Filtering*

In this section, the particle filtering is conducted to identify and track the user's palm in the region-on-interest reported in previous step. In condensation [7] algorithm, the probability distribution of possible interpretations is represented by a set of particles. Assume that the state of target object is modeled as *x* and the observation, the region-of-interest reported in the input image, is referred as $z$, we would like to compute the posterior density $p(x|z)$ indicating the probability the user's palm is relying on. A set of particles, denoted by $\left\{s_t{}^{(n)}, n = 1, \cdots, N\right\}$ with the weights of the particles $\pi_t{}^{(n)}$, is to express the posterior density $p(x_t|Z_t)$ at frame $t$.

The most important step in particle filtering is to compute the posterior probability of a particle. We combine three observations including depth, pose, and motion information of the user's hand to evaluate palm positions. Operating with QPalm, the user first rises his/her hand and then makes circles in the air. Based on the operation, the palm part of the arm has larger depth values than the rest and is usually positioned in the one of two ends of the hand shape. Combining the two observations, we can locate a palm position reliably. However in a living room scenario, there are possibly more than one people there. We should consider other people who may interfere with the operating user. For the

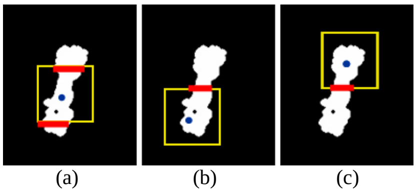


Fig. 5. Use the information of shape to reject the particle located on wrist or forearm. The right two images are particles on bad locations and the leftest one is a particle at good position.

people stand or walk behind the user, depth information can effectively ignore them. But for the people sit next to the user, depth information does not help much. To this case, we utilize the motion information of a palm to alleviate the kind of interference. The technique for evaluating a particle is listed as follows.

*1) Depth Information:* From last step, we have a probability map $P_t$ indicating the probability of the user's hand positioned. For each particle reporting a guess of palm position, say $(x, y)$, we compute the depth score, $M(p_{x,y}, \omega)$ as the average probability within a window centered on the position reported by the particle. The window size is set as a regular palm size in the image view. Since the palm part of the arm has larger depth values than the rest while operating with QPalm, we prefer the particle with higher depth score.

*2) Shape Information:* The connected components in ROI are taken part in the evaluation step as well by considering the features of a hand's shape. While performing circular and push motions, a palm is located on the top of the connected component. As shown in Fig.5(a), a good particle's window has intersections with the connected component at the bottom edge. A particle with intersections only at the upper edge of its window is filtered out since it is in the bottom of the connected component and cannot be a candidate of a palm (see Fig.5(b)). Also, if a particle is located on the forearm as shown in Fig.5(c), the connected component then intersects with the two opposite edges of the window. This information offers an excellent rejection filter for eliminating particle located on wrist or forearm.

*3) Motion Information:* The optical flow is applied to particle evaluation by considering the relationship between the location of a palm in the previous frame and the motion direction of the current frame (see Fig.6). Using optical flow makes the detection results more robust to the interferences in the environment. The flows within a window of a particle are first split to several clusters, $(c_1, n_1, \overrightarrow{v_1}), (c_2, n_2, \overrightarrow{v_2}), \cdots, (c_k, n_k, \overrightarrow{v_k})$, where $c_i$ means the $i^{th}$ cluster, $n_i$ is the number of optical flows in this cluster and $\overrightarrow{v_i}$ is the mean vector of this cluster. The vector $\overrightarrow{e_{p_{x,y}}}$ which formed by $l_{t-1}$, the previous location of a palm, and the location of the particle $p_{x,y}$ shows the potential direction of this particle. This vector should have a positive correlation with the optical flows nearby the particle. The correlation estimated function is shown in Eq.2.

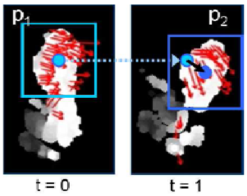


Fig. 6. Use the optical flow to examine the correlation between the previous location of the palm and the particles.

$$\mathrm{corr}(p_{x,y}) = \frac{1}{\mathrm{N}}\sum_{i=1}^{k}\frac{\overrightarrow{v_1}\cdot\overrightarrow{e_p}}{\|\overrightarrow{v_1}\|\|\overrightarrow{e_p}\|}\cdot n_i \tag{2}$$

By applying Eq.2 to particle evaluation, some moving objects near the user can be filtered out if they move in the different direction with the user. By combining all information, the evaluation of a particle is computed as:

$$conf(p_{x,y}) = \begin{cases} c_1 \cdot M(p_{x,y}, \omega) + c_2 \cdot corr(p_{x,y}), if\ s(p_{x,y}) = true \\ 0 \qquad\qquad ,if\ s(p_{x,y}) = false \end{cases}$$

$$s(p_{x,y}) = \textit{shape-detection}, \quad c_1 + c_2 = 1 \tag{3}$$

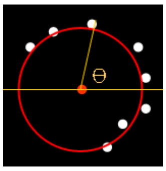
,where $M$ is a mean filter using ω as its window size. $c_1$ and $c_2$ are the weights which make $conf(p_{x,y})$ to be a linear combination of the mean filter and correlation. After applying Eq.3, the goodness of all particles is determined. In next frame, the best $n$ particles are selected and this process then iterates with new observations.

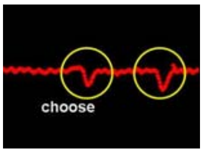
## IV. Motion Analysis

In previous section, the method for detecting and tracking user's palm is demonstrated; hence the trajectory formed by the location of a user's palm is extracted. Two kinds of trajectories recorded are the 2D positions of the palm and their variations in depth. The 2D trajectory is used to detect if it expresses a circle, which means the user is drawing a circle, and the trajectory in depth is employed to determine if a push motion is occurred.

### *A. Detection of Circular Motion*

A prediction of circular motion is required since the system should react during a motion, rather than after an entire circle. For detecting the circular motion, a best-fit first estimated based on the trajectory. Then, we check if the order of the points on the trajectory truly describes the circle or just a causal motion made by the user. Assume that we have the trajectory, named as $T_t = \{(x_t, y_t), (x_{t-1}, y_{t-1}), \ldots, (x_{t-k}, y_{t-k})\}$, and $k$ is the number of recorded points. An estimated circle is then computed by solving the





over-constrained system [5].

$$\begin{bmatrix} 2x_t & 2y_t & 1 \\ 2x_{t-1} & 2y_{t-1} & 1 \\ \vdots & \vdots & \vdots \\ 2x_{t-k} & 2y_{t-k} & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} {x_t}^2 + {y_t}^2 \\ {x_{t-1}}^2 + {y_{t-1}}^2 \\ \vdots \\ {x_{t-k}}^2 + {y_{t-k}}^2 \end{bmatrix} \qquad (4)$$

Eq.4 is a basic form of *Ax=B* where *x* can be solved by the least squares method. It minimizes the errors between observed data and the predictive model. In our system, the predictive model is a circle and the best-fit circle is then assumed to be centered at *($c_1$, $c_2$)* with radius $r_t = \sqrt{c_3 + {c_1}^2 + {c_2}^2}$.

When the best-fit circle is estimated, the next step is to determine if the trajectory truly expresses this circle. The first and intuitive information is that the mean square error between an observed point on trajectory and its nearest point on the predicted circle should be small. Moreover, while an user draws a circle, the palm location should be ordered in a clockwise or counterclockwise way. Hence, the angle which is formed by the palm location, the center of best-fit circle and the horizontal line should increase or decrease progressively (see Fig. 7(a)). This method effectively eliminates noises and casual motions. At last, a circle with too large or too small radius is removed since in most time it is formed by a straight line or a motionless gesture.

### *B. Push Motion Detection*

A push motion is quite different from the circular motion since the system only needs to react one time after it detects a legal motion. There are several related works aim at recognizing a pre-defined gesture and most of them adopt the approach which trains target motion at first. In our situation, a more straightforward way can be used to solve this problem because a push motion has clear changes in depth, as in Fig. 7(b). A circular motion is often performed parallel to the image plane of a camera, while a push motion is perpendicular to the image plane. Besides, a push motion must appear after a circular motion. As a result, we detect a push motion by checking if there is a continuous increase in palm's depth while the user is drawing a circle. If a push motion is detected, there is one second delay for accepting another push gesture. This technique prevents producing multiple choosing events during one choosing motion.

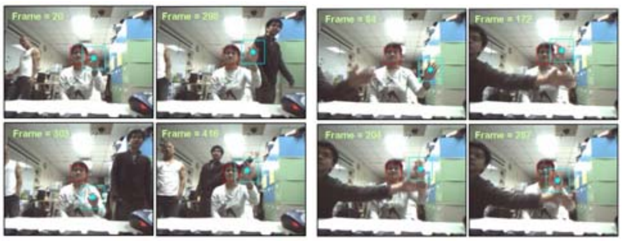


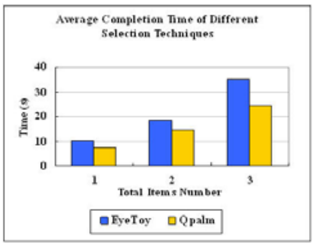
Fig. 8. (a) Some people passed by the user. (b) A person sit by the user and tried to interfere the user.

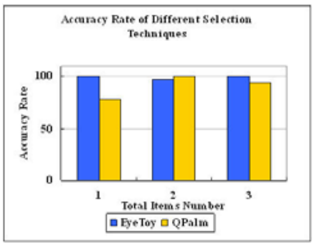
## V. EXPERIMENTS AND USER STUDY

The proposed method is applied to two videos which simulate the environment in a living room. The first video has 476 frames while the second video has 325 frames, and both with a 320 x 240 resolution. It runs on a PC with Intel Pentium 3.40G Hz CPU plus 2.49GB RAM and all tasks were implemented by C++ code. The stereo camera is equipped with two 2.8 mm lens on it. The distance between camera and users is about 2 m and the frame rate is 16 to 18 frames per second (fps). After applying our algorithm, the frame rate descends to 10 to 12 fps. Some results are shown in Fig.8.

In video 1, a person passes by the user at frame 20 and 298. There are also multiple passers appeared at frame 303 and 416. The detecting result is not affected by the passers since the depth information is used in this system. Additionally, in video 2 a person tries to influence the user by making some movements nearby the user at frame 84, 172, 204 and 287. The detecting results show that the system is robust to the interference from other people.

**User Study:** In order to check the practicability of the interface QPalm, a series of experiments and comparisons were taken. In these experiments, participants were asked to select a specified item in a list menu with two kinds of selection techniques: the EyeToy-like selection technique and QPalm. In EyeToy's design, a camera is set in front of the users and they can see their own image on the display as a feedback. A previous, a next and a choosing button are placed on the display and users simply touch and wave in the area of these buttons to make a command to the menu. An experiment contains three blocks and each corresponds to those two different techniques. What participants see is a simple list menu and the items are showed vertically (see Fig.1). The target item's number is





generated randomly in the range of menu size. Before one block starts, we teach users the technique allowed in next block and let users scroll on the list menu until they satisfied. Besides, we put three practical trials in the beginning of a block to ensure the participant getting used to the specified technique. In a block, we ask users to repeat five tasks in different menu sizes (20, 50 and 120) hence there are 2 * (3 + 5 * 3) = 36 trials in an experiment. A total of 8 participants took part in the study and comprised of 4 males and 4 females. Half of them had the experience of playing in Sony's EyeToy game while all of them were skilled in using computer. Another, there were no users familiar with the use of QPalm.

**Completion Time:** We reported the result of completion time and showed it in Fig. 9(a). Note that we can directly see a significant variation while the total item number is increased. Also, the EyeToy's technique performs a longer completion time than the other two techniques in all these trials. While the total item number is 20, there are no obvious differences in these two techniques. However, if the total number increased to 120 (35.077s), the EyeToy's method showed a worse result than QPalm (23.933s). Users responded that they can quickly scroll to their concerned area by QPalm techniques but cannot by an EyeToy-like interface.

**Accuracy:** In these tests EyeToy's technique shows a better result since it allows user to choose the desired target more slowly. There is a worse outcome (83.76\%) in QPalm while total item number is 20. It is because that we let users judge if they were ready for the experiment themselves. Users may not actually be proficient in the selection technique but enter a true experiment. The results of QPalm technique after the first session are obviously better when total items number increases.

**Questionnaire Responses:** Most participants reported that a gesture-base method helps them jump to their concerned region faster. In order to examine the relationship between total item number and these two techniques, we asked user to subjectively choose which one they preferred under different total menu size (20, 50 and 120). According to Questionnaire responses, users preferred the technique. However, if the number of items is small, users considered the EyeToy-like technique more easily to perform.

## VI. CONCLUSION

The proposed a novel interaction technique, QPalm, which allows the user to control the media via a list menu shown on a distant display by drawing circles in the air with one hand. QPalm provides two basic functions, circular and push motions, for the user by recognizing the user's palm trajectory in three-dimensions. The circular motion provides fluidity in scrolling a menu up and down, while push motion is intuitive when the user decided to choose an item during a circular motion. In the future, we would like to extend QPalm in a multiple-user scenario. While people are watching television, it is usual that more than one person are trying to control the menu. It is impossible that the system gives the control of menu to all users at a time. A possible solution is to set two cameras in the environment, one with wide-angle lens and the other with pan-tilt-zoom function. When the system detects a motion asking for the control in the wide-angle camera, the pan-tilt-zoom camera then focuses on the interest area.

## ACKNOWLEDGMENT

This work was supported in part by the National Science Council, Taiwan, under grants NSC 96-2752-E-002-007-PAE, and by the Excellent Research Projects of National Taiwan University, under grant 95R0062-AE00-02.

## REFERENCES

[1] William T. F., Craig D. W.: Television Control by Hand Gestures. IEEE Intl. Wkshp. on Automatic Face and Gesture Recognition. (1995) 179-183

[2] Tomer M., John F. H.: Navigating documents with the virtual scroll ring. Symposium on User Interface Software and Technology. (2004) 57-60

[3] Smith G. and Schraefel M. C.: The Radial Scroll Tool: Scrolling Support for Stylus-or-Touch-Based Document Navigation. Proceedings of the 17th annual ACM symposium on User interface software and technology (2004) 1:53-56

[4] Grham, S., Schraefel M. C., Patrick B.: Curve dial: eyes-free parameter entry for GUIs. Conference on Human Factors in Computing Systems, CHI. (2005) 1146-1147

[5] James A., Kevin N.: Fluid sketches: continuous recognition and morphing of simple hand-drawn shapes. Proceedings on User interface software and technology. (2000) 73-80

[6] Ivan L., Tony L.: Tracking of multi-state hand models using particle filtering and a hierarchy of multi-scale image features. Proceedings of the Third International Conference on Scale-Space and Morphology in Computer Vision. (2001) 63-74

[7] Michael I., Andrew B.: CONDENSATION conditional density propagation for visual tracking. International Journal of Computer Vision. (1998) 29,1:5-28.

[8] Rainer L. and Jochen M.: An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP. (2002) 1:900-903

[9] Dorin C., Visvanathan R., Peter M.: Real-Time Tracking of Non-Rigid Objects using Mean Shift. Computer Vision and Pattern Recognition. (2000) 2:142-149