HAMERA: a Device for Hand Profile Construction in the Pervasive Environment

Xun Luo
Office of the Chief Scientist R&D
Qualcomm Inc.
San Diego, CA 92121
Email: xun.luo@ieee.org

Robert V. Kenyon
Electronic Visualization Laboratory
University of Illinois at Chicago
Chicago, Illinois 60607
Email: kenyon@uic.edu

Fig. 1. Font and back views of a prototypical HAMERA device.

***Abstract*—This paper presents the design and implementation of HAMERA (Hand cAMERA), a novel device for hand profile construction in the pervasive environment. With the help of software, HAMERA requires the minimum hardware of a mainstream mobile device enhanced with a single accelerometer, and is capable of providing advantageous features including high-quality hand image collection, on-device profile construction and easy usability. Evaluation results show the efficiency of HAMERA in serving its goals.**

I. Introduction

One of the ultimate goals of pervasive computing is that the user is able to interact with computer systems at anywhere and anytime. Vision-based gesture interface is a promising technology to help achieving this goal by using the most dexterous part of the human body, i.e. the hand, as a natural human-computer interface(HCI). Encouraged by this potential, vision-based gesture HCI has been an active research topic for the past two decades [1], [2]. To name a few representative systems, Freeman et al [3] used a video camera to detect hand movement. By waving the hand from side to side, the user was able to remotely control the sound of a television. Segen et al constructed a two-camera system to extract fingertips of the hand, applied reverse kinematics to drive an articulated hand model, and subsequently used the derived hand model for gesture-commands interpretation [4]. More recently, Kolsh et al developed HandVu [5], a vision-based gesture HCI which was designed primarily for wearable systems. In its most common configuration where the camera is fixed on a head-mounted display (HMD), HandVu is capable of working under different illumination conditions as well as with various image backgrounds. With the advances of computer vision research and CPU processing power, it is reasonable to expect that the number of gesture HCIs will continue to grow.

People's hand appearances distinct from each other, and using an one-size-fit-all hand profile for everyone will not be sufficient. Figure 2 shows the hand images we collected from two subjects: one southwest Asian and one southeast Asian. The appearance differences can be clearly observed even though collection conditions have been carefully calibrated to be identical for both subjects. For accurate identification of a specific user's hand, it is necessary for the gesture HCI to have the user's personalized hand profile. Once the profile is acquired, a gesture HCI in the pervasive environment can work in a way similar to the "Personal Augmented Computing Environment" [6], [8] paradigm or "Carry Small, Live Large" scheme [7]. In such scenarios, personalized hand profiles can be stored on a portable device carried with the user. When interaction with a computer in the pervasive environment is taking place, the personalized hand profile is retrieved and subsequently used by the gesture HCI.

Construction of personalized hand profile in the pervasive environment is not a trivial research problem. This process requires the user herself, who is usually not a professional photographer, to collect images of the hand of high quality and considerable quantity. It also involves post-processing of the collected hand images for feature extraction and model training. Lastly, both the image collection and post-processing are ideally to be accomplished by one single device with small form factor, low-cost hardware, low-power consumption and easy-to-use user interface. To the best of our knowledge, there is not such a device yet in the industry or academia that satisfyingly offering the characteristics needed. This paper presents HAMERA (Hand cAMERA) to address this gap. HAMERA is an enhanced personal mobile device for hand profile construction in the pervasive environment. It stands out by making three major contributions. First, its minimum hardware requirement is easy to satisfy, which is a mobile device that: 1) has a platform (e.g. Windows Mobile, Linux, Android or Java) to run third-party applications. 2) has a camera component. 3) is equipped with a single accelerometer. Most of the contemporary mobile devices have 1 and 2 ready. Accelerometers are not prevalent on mobile devices yet but

978-1-4244-3304-9/09/$25.00 ©2009 IEEE

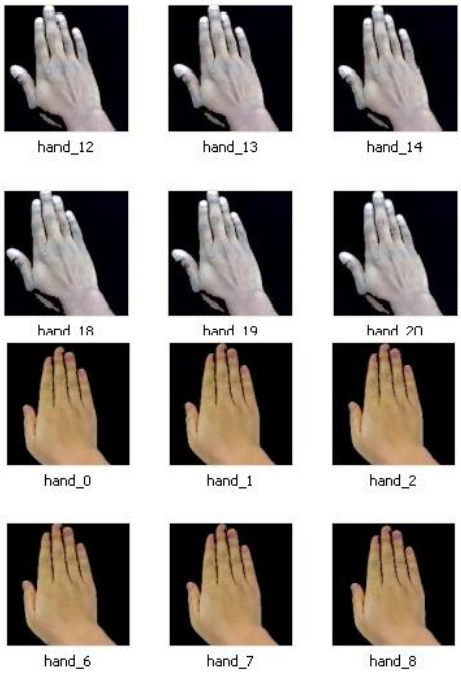

Fig. 2. Hand images from a southwest Asian subject (top six) and a southeast Asian subject (bottom six).

the addition of them is cost and energy efficient. Secondly, image quality control is done automatically without the need for manual intervention during collection. The user simply sets collection parameters in a configuration UI once and monitors a viewfinder UI for the rest of the collection process. Lastly, post-processing of collected images is done on the device and thus the constructed hand profile is portable with the user.

This paper is organized as follows. After a review of existing technologies in the literature in Section II, Section III presents the hardware construction of HAMERA. The software implementations are described in Section IV and evaluation results are provided in Section V. Section VI concludes the paper and points out several possible future directions.

II. Related Work

The state-of-art approaches to construct a hand profile can be summarized as a process with two steps:

1) Collecting the hand images of a user in a controlled environment. The environment is usually a controlled setting with fix-mounted cameras and light sources.
2) Processing the hand images collected into a hand profile. Under most cases, the computer that does post-processing is a separate device from the image-capturing device, which is the camera.

To undertake step 1 in the pervasive environment is a well-known challenge because of three sources of image noise: illumination fluctuation, image blur and hand depth variation. Illumination fluctuation, caused by the lighting condition change in the environment, results in intensity discrepancy across images. Image blur, caused by the camera or hand movement, results in reduced sharpness of the images. Hand depth variation, caused by the change of distance between camera and hand, results in inconsistent size ratio of the hand against the background. All these three phenomena undermine quality of the hand profile constructed and need to be properly handled. To deal with illumination fluctuation, contemporary digital cameras (DCs) usually use automatic exposure control across whole image. Automatic exposure control calculates the overall intensity of the captured frame, and adjust this value to be uniform across images. For image blur, different DCs use different image stabilization (IS) techniques, including optical IS, mechanical IS and digital IS. Optical IS captures the camera movements with the help of sensing components embedded in the lens, and adjusts the lens accordingly to offset these movements; mechanical IS compensates camera movement by maneuvering the image sensor; digital IS corrects the captured image in a post-processing fashion to eliminate the motion artifacts. To handle hand depth variation, the best way nowadays is to instruct the user to hold the camera steadily and maintain a constant distance between the camera and the hand.

It can be seen that the existing solutions that deal with the three phenomena have shortcomings. For illumination fluctuation, automatic exposure control can balance the intensity across images, but is a naive solution that does not distinguish the hand and the background. As image blur is concerned, optical IS requires the incorporation of delicate sensing components in the lens, for devices that use low-cost lens this is usually not feasible. Because mechanical IS requires maneuvering of the image sensor for movement compensation, it brings undesired higher cost to the device. Digital IS is a passive method and can not fully eliminate motion artifact. Letting the user estimate and maintain the camera-hand distance to deal with hand depth variation requires the user to have good manual skills and is not very realistic. HAMERA takes different paths to address these phenomena. To fight illumination fluctuation, it employs foreground segmentation to identify the hand blob in the whole image, and then controls the image intensity only within this blob. The advantages of this approach is that it is independent of intensity noise in the background, and the exposure uniformity across multiple hand images are ensured. In order to avoid image blur, HAMERA actively monitors the camera shaking and tilt angle with the accelerometer, and only collects images when these two measurements are below certain thresholds. In comparison with the passive strategies that compensate camera movement, this strategy is preventive and reduces the image blur by attacking its root cause. To deal with hand/backgroun variation, HAMERA calculates the foreground/background pixel ratio to make sure that this value is within the allowed range, otherwise images are not collected. As shown in Section V, this is a very effective method in controlling hand depth variation and puts no burden on the user for maintaining camera-hand distance.

In addition to its enhancement to step 1, HAMERA also has another advantage by performing a large portion of step

2 on the device itself. For the collected raw hand images which are often in the quantity of dozens or even hundreds, HAMERA conducts post-processing on each of them. This post-processing is composed of several stages. The first stage is foreground segmentation, where the pixels belonging to the hand are separated from the background pixels and the background is subsequently replaced with a blank canvas. The second stage is exposure adjustment on the segmented hand image, which makes image intensities consistent across all hand images. During this stage, the illumination conditions of different raw hand images are normalized. In the third stage HAMERA extracts the skin color information and builds a color histogram to represent the user's skin tone. This histogram-form color model is stored as part of the user's hand profile. In the last stage, HAMERA converts all the hand images into a set of binary format images, i.e. black-and-white templates. This set of templates is used to construct another key part of the hand profile – the shape model. It is worth noting that HAMERA still relies on external computers to aid the shape model construction, which is usually a very computationally intensive task.

III. Hardware Construction

Figure 1 shows a prototypical HAMERA device we built, with a Motorola A1200 smart phone being the baseline mobile device. As previously stated, HAMERA can be built over any mobile device that allows third party application deployment and has a camera component. Our choice of A1200 is based solely on its availability.

The A1200 smart phone runs MontaVista Mobile Linux operating system and supports third party applications in the form of Linux executables or Java midlets. Processing unit of the A1200 is an Intel XScale 312MHz processor which is compliant with ARM micro architecture. The camera component consists of a 2 Megapixel Micron MT9D112 system-on-a-chip sensor and a micro lens. Three resolution are provided, which are $1200 \times 1600, 768 \times 1024$ and 480×640 respectively. The camera component supports both normal and micro focusing modes. The A1200 device has an extensible storage capacity of up to 2 Gigabytes. In the prototypical configuration a 1 Gigabyte micro SD card is used. It is also equipped with a 2.4 inch touch screen with 320×240 resolution.

An wireless accelerometer board is mounted on the back (battery cover) of the A1200 device to augment it with acceleration sensing ability. The sensor used is a STMicro LIS3LV02DQ digital MEMS accelerometer, which can measure 3-axis acceleration data. The LIS3LV02DQ has a user selectable full scale of $\pm 2g, \pm 6g$ and it is capable of measuring acceleration over a bandwidth of 640 Hz for all axes. Measurement outputs of the accelerometer is transmitted wirelessly through Bluetooth link to the A1200 device. The Bluetooth transceiver used here is CSR BlueCore3, a system-on-a-chip with micro controller and a DSP co-processor. The accelerometer and Bluetooth transceiver are co-located on the accelerometer board which is self-powered with a chargeable battery pack.

Fig. 3. Raw hand image from the camera video feed (left), the binary mask created by shresholding (middle) and foreground-segmented hand image (right).

IV. Software Implementation

The software component of HAMERA orchestrates the collaboration among HAMERA hardware modules. It consists of four modules: the configuration UI module, the accelerometer integration module, the viewfinder UI module and the image post-processing module. When the user uses HAMERA for hand profile construction, these four modules work in a process as shown below:

- The user sets up image collection parameters using the functionalities provided by the configuration UI module. These parameters include number of images to collect, collection interval, foreground segmentation threshold and hand depth range.
- While the camera component of HAMERA is capturing a live video feed of the user's hand, the accelerometer integration module works in parallel and processes readings from the accelerometer. The acceleration readings are converted into two measurements of the camera: shaking motion and tilt angle.
- The viewfinder UI renders each frame of the live video feed on the HAMERA screen. If the shaking motion and/or tilt angle are above allowed limits, then the corresponding frames are not further processed and subsequently discarded. Otherwise, the viewfinder UI performs foreground segmentation to separate the hand from its background, as shown in Figure 3.
- The depth of the hand is derived by comparing the segmented foreground with the whole image. If this depth is within the range defined in the configuration UI module, the frame is further processed otherwise it is discarded.
- If a frame is captured under satisfying shaking motion, tilt angle and hand depth conditions, it is a qualified hand image and stored for hand profile construction. The first qualified hand image's foreground intensity value is used as a reference for the illumination control of subsequent hand images.
- Once the specified number of hand images are collected, these hand images are processed by the image post-processing module to construct the hand profile, including a skin color model and a hand shape model.

In the following sections the four software modules are described in detail.

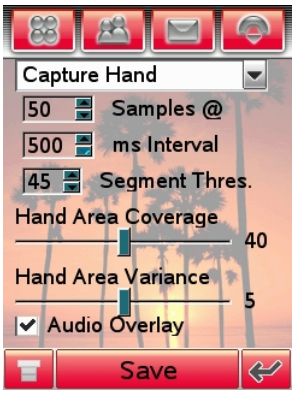

Fig. 4. Screenshot of the configuration UI on HAMERA.

Fig. 5. Screenshot of the viewfinder UI on HAMERA.

A. Configuration UI Module

Figure 4 shows a snapshot of the configuration UI component. This component allows the user to set six parameters for hand image collection. *Number of samples* specifies how many hand images are to be collected in total during a session. *Collection interval* is the minimum time span that needs to elapse between taking two consecutive hand images. HAMERA assumes that during hand image collection, the background is always at a lower intensity than the hand, so *segmentation threshold* is the intensity threshold used to segment the hand (as foreground) out of its background. It is represented as the percentage of the highest pixel intensity value in the image. *Hand area coverage* and *hand area variance* describes the user-defined percentage of hand pixels in the whole image. These two parameters together set the hand depth range. *Audio overlay* configures that if there is audio feedback during hand image collection. By default the feedback is visual and displayed by the viewfinder UI module. If this option is turned on then text-to-speech audio feedback is enabled in the viewfinder UI component.

B. Accelerometer Integration Module

The accelerometer integration module is responsible for converting raw data from the HAMERA accelerometer into two metrics: shaking motion and tilt angle. HAMERA senses the device acceleration along the three local coordinate axis with regard to the accelerometer. The overall acceleration a, calculated in Equation 1 as the rooted square sum of the three axis-wise readings, can be used directly as the shaking motion metric, represented in the quantities of unit gravity acceleration:

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{1}$$

When the device's shaking motion is at low level, its tilt angle is also reported by the accelerometer integration module. The accelerometer is defined to reach its steady state conditions in the state of rest or uniform motion characterized with 1g acceleration caused by gravity. In the context of HAMERA, steady state is defined as the state of rest. The accelerometer in its steady state satisfied Equation 2 and can be directly used to measure the gravity vector g which is always vertical to the horizontal plane. Tilt angles are then calculated from three orthogonal acceleration components as shown by Equation 3 and 4:

$$a \simeq g \tag{2}$$

$$\Theta = \arcsin(\frac{a_x}{g}) \tag{3}$$

$$\Phi = \arcsin(\frac{a_y}{g \times \cos\Theta}) \tag{4}$$

The vector $[a_x\ a_y\ a_z]^T$ represents the gravity vector measured in its local coordinate axis, Θ and Φ are pitch and roll angles in the global coordinate axis.

C. Viewfinder UI Module

The task of the viewfinder UI module is to provide the user with visual and optional audio feedback of the hand image collection progress, as well as store the qualified hand images for the image post-processing module. It displays the video stream from the camera component on the HAMERA screen, together with other informational statu messages. Figure 5 illustrates a snapshot of the viewfinder UI component. It can be seen that the screen area is divided into three regions. The top one is the video feed displaying region, where the user can see the live video feed from the camera component. Below the video feed region is the status messages region. This region displays several pieces of information, including the parameter values set by the configuration UI module, magnitude of shaking motion, the hand depth factor, and reminders for the user to make camera adjustments. At the bottom is the hand image counter region, where the number of total hand images to be collected, as well as qualified hand images that have been stored are shown.

Two image related metrics are calculated by the viewfinder UI component online. They are the illumination factor IF_{hand} and the hand depth factor DF_{hand}. As indicated by Equation 5, illumination factor is the average intensity value across all the pixels in the segmented hand foreground. For the hand

images of a specific user, higher IF_{hand} implies brighter illumination, while lower IF_{hand} means the contrary. The hand depth factor is calculated using Equation 6. It is the area percentage of segmented hand foreground against the whole image. Because that for a specific user the physical hand size is an invariant, higher DF_{hand} maps to closer hand depth, and lower DF_{hand} indicates farther hand depth.

$$IF_{hand} = Avg(\Sigma(I_{hand}(i))) \tag{5}$$

$$DF_{hand} = \frac{P_{hand}}{P_{image}} \times 100\% \tag{6}$$

Even if the collection conditions for a video frame satisfy the requirements for camera shaking motion and tilt angle, that video frame still needs to be checked for its illumination factor and hand depth factor values. Only when these two values conform to the user defined ranges the video frame is determined to be a qualified hand image. To be more specific, the illumination factor has to be within at most $\pm 5\%$ off the illumination factor of the first qualified hand image (the only hand image that does not need to be examined for its illumination factor). At the same time, the hand depth factor needs to be within (*hand area coverage*) $\pm$ (*hand area variance*), as provided by the configuration UI module.

D. Image Post-processing Module

The image post-processing module fulfils the task of constructing hand color and shape models, which are the two elements of a personalized hand profile. This module does not have a UI. All computation is performed by a background process and does not affect the other usages of the HAMERA device.

Firstly the background of the hand images stored by the viewfinder module are removed. The background pixels are replaced with zero-intensity ones to make the hand foreground appear to be taken in front of a blank canvas. The following step is hand image intensity normalization. The image post-processing module calculates the average intensity values across all hand images, and then reward the under-illuminated images by lifting its intensity and punish the over-illuminated images by lowering its intensity. This step balances the illumination factors of the hand images.

The post-processing module then constructs the hand color model. This model is represented in the form of a color histogram. The RGB color space is divided into a group of bins. In our prototypical device, the number of bins is 2^{16}. This is equivalent to 65536 color bins. Because the camera component of the Motorola A1200 represents each pixel with an 18-bit RGB value, each pixel's color value is hashed to map to a 16 bit bin. After all pixels of the collected hand images have been hashed into bins, numbers in the bins are normalized to be in the [0,1] range, i.e. percentages of the total number of pixels. The hand color model is subsequently persisted as part of the personalized hand profile.

Another part of the hand profile – the shape model – is constructed as follows. Color is not needed in this stage so the hand images are converted from RGB format to binary format with each foreground pixel set at intensity 1 (background pixel are of intensity 0 after the background removal step). The image post-processing module then packs all the hand images into a training data set. This training data set can be downloaded through wired or wireless link to a more powerful computer to construct the hand shape model. After the external computer finishes shape model construction it sends back the model to HAMERA for on-device persistence. The reason that an external computer still needs to participate is because that shape feature extraction and model training are usually computationally intensive, for example with Viola-Jones [9] or Kanade-Lucas-Tomasi [10] features. However, the image quality control and post-processing work performed by HAMERA makes these work much easier and more efficient for the external computer.

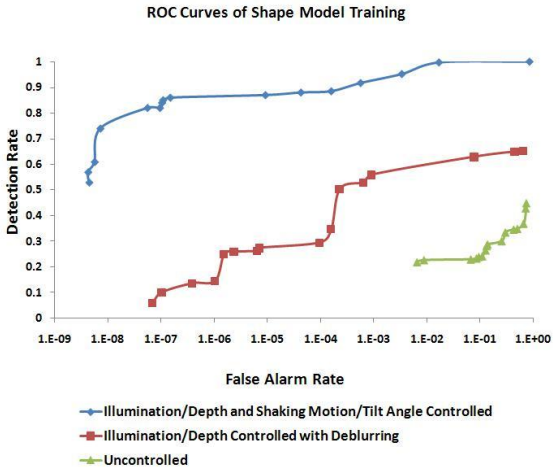

Fig. 6. ROC curves showing the contribution of HAMERA to hand profile quality.

V. Evaluation Results

To evaluate HAMERA, we implement a gesture HCI using techniques similar to those of the Intel face tracker [11]: hand and non-hand images are collected to train the hand shape models; Trained hand model are stored as posture templates; A correct recognition of the match between the posture template and a sub-region of the test image is counted as a detection, while an incorrect match is counted as a false alarm. Each hand model is trained with 500 hand images and 4000 non-hand images. The first 50 hand image are collected and the remaining 450 are derived applying a random ± 5-degree rotation on the collected images. Three collection conditions in the pervasive environment are tested. The first condition enables HAMERA's illumination/depth control as well as shaking motion/tilt angle control. The second condition keeps the illumination/depth control of HAMERA but relaxes the shaking motion/tilt angle control. Instead, it does deblur processing on the collected hand images using Lucy-Richardson algorithm [12], a prevalent digital IS method. The third condition does not enforce any controls during the hand image collection.

The 500 hand and 1500 background images are then used to train the Viola-Jones [9] type shape models. This is done for the three hand image collection conditions. The remaining 2500 background images are used as testing images to derive detection rate and false alarm rate data. These data are plotted as curves in the receiver operating characteristics (ROC) graph. Each ROC curve consists of 16 data points of detection rate/false alarm pairs, which are obtained by varying parameter configuration of the corresponding hand shape models.

Figure 6 shows the ROC curves for the hand shape models trained under the three conditions. It can be observed that without any constraints enforced, noises in the training data set result in expected low-quality shape models: the best detection rate is 0.43, while the best false alarm rate is $6.44 * 10^{-3}$. Given that the template size is of 25 $\times$ 25 pixels and the test image size is of 640 $\times$ 480 pixels, each test image contains about $4.7 * 10^4$ sub-areas at the size of the template. The false alarm rate is consequently equivalent to 8-9 false alarms per test image. This accuracy performance is far from acceptable to be used by a practical gesture HCI. Applying illumination/depth constraints significantly improves the performance of the corresponding hand shape models, the best false alarm rate improves to $7.02 * 10^{-8}$, this is about one false alarm every 300 test images when performing 25 $\times$ 25 size template scan over 640 $\times$ 480 test image. However, when the best false alarm rate is achieved the detection rate is still at a low level of 0.06. Actually, the best detection rate of the trained hand shape model is 0.65. Obviously illumination/depth constraints alone do not guarantee high-quality hand shape models, albeit hand models trained under this condition largely outperforms the counterparts under uncontrolled method. When both illumination/depth and shaking motion/tilt angle constraints are enforced, the trained hand models are able to archive both satisfying detection and false alarm rates. When the detection rate is 0.82, the false alarm rate is $5.6 * 10^{-8}$, which is about one false alarm every 380 test images. In a real life vision-based gesture HCI, when the camera captures 640 $\times$ 480 frames at 50 Hz, this class of hand shape model quality is equivalent to one false alarm in every 8 seconds and can be regarded as of high grade.

VI. Conclusion

In this paper the design and implementation of the HAMERA device are described in detail. The main contribution of this device is that it enables high-quality and efficient hand profile construction in the pervasive environment, which to the best of our knowledge is not yet a well-solved problem. HAMERA shows that accelerometer can be effectively utilized to address an important challenge – the image blur caused by camera shaking motion and tilt angle. It also proves that dynamic image features are helpful in the controlling of good illumination and depth conditions. The evaluate results justifies HAMERA's capability for its design purpose.

There are several directions of future work. One is further investigation of a broader range of sensors to benefit the process of hand profile construction. Besides accelerometer, several other sensors can also improve the image collection performance in multiple aspects. For example, a magnetometer can measure the global orientation of the user, and is thus helpful in illumination tuning when natural lighting is present. Similarly, a gyroscope attached to the user's wrist can report the wrist's angular motion. This motion data could further assist the identification of the camera tilt angle in multiple directions. There is a large free space to explore the proactive use and fusion of sensor data on mobile device.

Acknowledgment

We would like to thank Weimin Xiao and Magdi Mohamed from Motorola Labs for many useful discussions.

References

[1] H. Kage, K. Tanaka, K. Kyuma, C. D. Weissman, W. T. Freeman, W. T. Freeman, P. A. Beardsley, and P. A. Beardsley, "Computer vision for computer interaction," *ACM SIGGRAPH Computer Graphics*, vol. 33, pp. 65–68, 1999.

[2] J. L. Crowley, J. Coutaz, and F. Berard, "Perceptual user interfaces: things that see," *ACM Communications*, vol. 43, no. 3, pp. 54–58, 2000.

[3] W. T. Freeman and C. D. Weissman, "Television control by hand gestures," in *Proceedings of International Workshop on Automatic Face and Gesture Recognition*, June 1995, pp. 179–183.

[4] J. Segen and S. Kumar, "Look ma, no mouse!" *ACM Communications*, vol. 43, no. 7, pp. 102–109, 2000.

[5] M. Kolsch, M. Turk, T. Hllerer, and J. Chainey, "Vision-based interfaces for mobility," in *In Intl. Conference on Mobile and Ubiquitous Systems (MobiQuitous)*, 2004.

[6] X. Luo, "Pace: Augmenting personal mobile devices with scalable computing," in *Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid 2007*, Rio de Janeiro, Brazil, May 2007, pp. 875–880.

[7] R. Want, "Carry small, live large," *IEEE Pervasive Computing*, vol. 6, no. 3, pp. 2–4, 2007.

[8] X. Luo, *Personal Augmented Computing Environment: a Framework for Personalized Visualization and Scalable Human-Computer Interaction*. VDM Verlag, 2008.

[9] P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.

[10] J. Shi and C. Tomasi, "Good features to track," in *CVPR '94: Proceedigns of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994*, Jun 1994, pp. 593–600.

[11] Intel, "Computer vision face tracking for use in a perceptual user interface," Intel Technology Journal, http://developer.intel.com/technology/itj/q21998/articles/art_2.htm, Q2 1998.

[12] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *The Astronomical Journal*, vol. 79, pp. 745–753, Jun. 1974.

[13] J. Joseph and J. LaViola, "A survey of hand posture and gesture recognition techniques and technology," Providence, RI, USA, Tech. Rep., 1999.

[14] C. Reiter, "With j: Fast fourier transforms and removing motion blur," *SIGAPL APL Quote Quad*, vol. 31, no. 1, pp. 16–17, 2000.