

UCC-UFC: An Efficient Application of User Created Content in Ubiquitous Fashionable Computer

Yo-Won Jeong, Chul Lee, Ki-Woong Park, and Kyu Ho Park

Abstract—Comparing with conventional mobile devices, the major attractions of the wearable computer are high mobility and convenient usability by its wearability. The one of important issues for the success of wearable computer is developing its killer applications. An application of user created content (UCC) in Ubiquitous Fashionable Computer (UFC) [1], introduced in this paper, is an efficient and useful tool to create and manage user's multimedia content. The UFC is a wearable computer developed for obtaining high wearability, usability, aesthetic appearance, and social acceptance. The UFC includes a gesture-based user interface called iThrow. By using this device, users can control UFC easily and intuitively. The scenario of our UCC in the UFC is as follows. First a user captures an image by using the camera attached in the UFC as shown in Fig. 1. For editing the captured image, the user can send it to a near kiosk by taking throwing action to the kiosk, and then the image is displayed in the kiosk. The user can edit the image such as rotation or adjusting picture size, brightness, and contrast by taking various intuitive hand-gestures. After editing, if the user takes the throwing action again, the final image is uploaded and registered in his blog automatically. Therefore, if users follow this scenario, they can fast and easily enjoy UCC services.

I. INTRODUCTION

In recent year, wearable computer and ubiquitous computing system environment have been realized due to the rapid progress of computing and communication technology. Wearable computer can be broadly defined as mobile electronic devices that can be unobtrusively embedded as part of garment or accessory [2]. Comparing with conventional mobile devices, the major attractions of the wearable computer are higher mobility and more convenient usability by its wearability. The wearable computer is always active, and running without user's attention and gives services to the user with the support of ubiquitous environment. In the ubiquitous environment, using the wearable computer, we can easily obtain necessary information anytime and anywhere with a minimal constraint.

The one of important issues for the success of wearable computer is developing its killer applications. It is desirable that the application currently widely used is improved to play the merits of the wearable computer such as its mobility and usability. In this paper, we present the design and implementation of a killer application for wearable computer. It is the efficient and useful tool to create and manage user's multimedia content called user created contest (UCC).

A famous columnist Thomas L. Friedman said "We are no longer consumers but producers as well". This means that now the users of online tools can also contribute information as

well and access it which has revolutionized the whole way we thought about the Internet, as previously it was referred to as a source of information and not something where one could contribute as well. Users can now create their own content online i.e. contribute information to the web and download as well [3]. Blogs are, of course, the biggest example for showing UCC is a killer application in ubiquitous computing environment.

However, users must put up with considerable inconvenience creating and managing their content in the conventional UCC. First they take a picture by using a digital camera or cell phone camera. Before upload the picture to their blog site, they must edit the picture such as rotation or adjusting picture size, brightness, and contrast. For that users must learn the usage of an image editing software such as PhotoShop[®]. After editing the picture, they can upload and register the picture by a web browser. Since the work of editing and uploading the picture must be performed in a desktop computer, users must move to the place where there is the desktop computer after taking the picture. These problems can be addressed by using the wearable computer and ubiquitous computing system environment.

In this paper, we present the design and implementation of an efficient and useful application for UCC in wearable computer. As the wearable computer, we used the ubiquitous fashionable computer (UFC) that our research group has developed by funding of Korean government for obtaining high wearability, usability, aesthetic appearance, and social acceptance [1]. Specially, because it is desired to develop comfortable and user-friendly input device, the UFC includes a gesture-based user interface called iThrow. By using this device, users can control UFC easily and intuitively. Also we have implemented kiosk for the ubiquitous environment. The kiosk is a kind of a public PC distributed widely in the public places. The public computer for book searching in the bookstore is an example of the kiosk. Our research group has implemented the kiosk interoperating with UFC.

The scenario of our UCC in the UFC is as follows. First a user captures an image by using the camera attached in the UFC as shown in Fig. 1. For editing the captured image, the user can send it to a near kiosk by taking throwing action to the kiosk, and then the image is displayed in the kiosk. The user can edit the image such as rotation or adjusting picture size, brightness, and contrast by taking various intuitive hand-gestures. After editing, if the user takes the throwing action

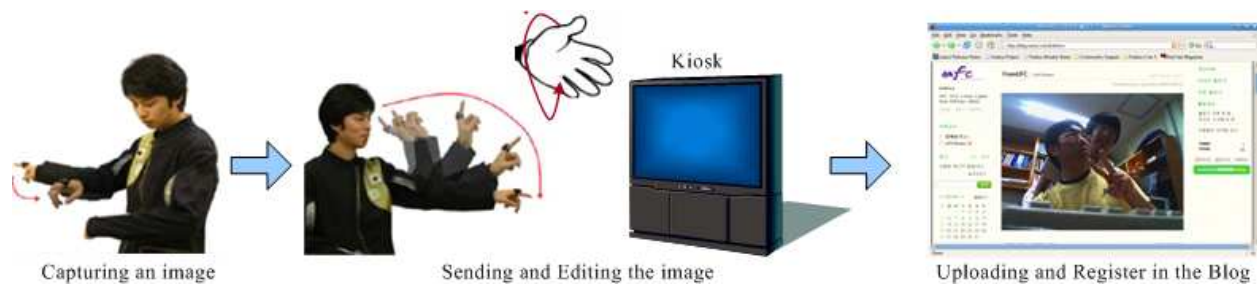


Fig. 1. The scenario of our UCC in the UFC

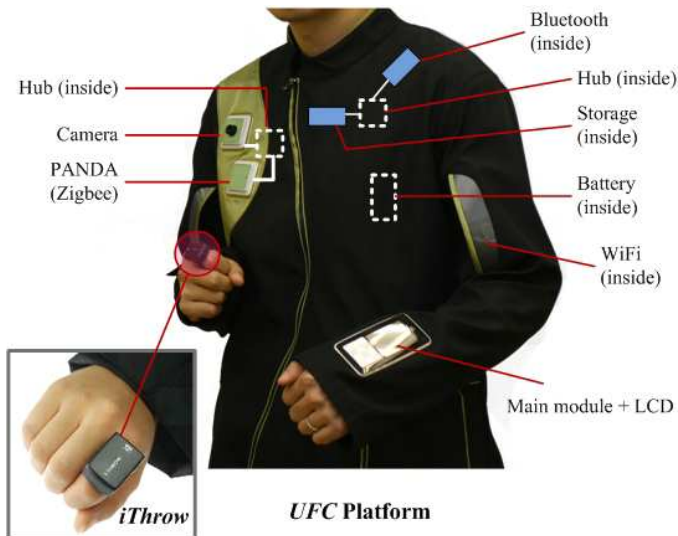


Fig. 2. UFC platform and iThrow user interface

again, the final image is uploaded and registered in his blog automatically. Therefore, if users follow this scenario, they can fast and easily enjoy UCC services.

The remainder of this paper is organized as follows. Section II describes the UFC platform with iThrow user input device. The design and implementation of our UCC using the UFC with iThrow is presented in Section III. Concluding remarks are presented in Section IV.

II. UFC WITH ITHROW

In this paper, we present the design and implementation of an efficient and useful application for UCC in wearable computer. As the wearable computer, we used the ubiquitous fashionable computer (UFC) that our research group has developed for obtaining high wearability, usability, aesthetic appearance, and social acceptance [1]. UFC platform is shown in Fig. 2. The UFC system consists of a portable wearable main module with various communication interfaces, user interfaces, and ubiquitous environment. In the main module, the core CPU is an ARM based Intel XScale-PXA270[®] processor having 624MHz and the main memory is 256MB Samsung mobile-SDRAM[®] having 104MHz. Other UFC modules are distributed on a garment, considering the distribution of weight and aesthetic design. Since we utilized

a standard USB protocol to communicate between the main module and other UFC modules, due to the hotswap capability of USB devices, each UFC module can be attached and detached easily on a garment while the system is running [4]. Operating system running on UFC is GNU/embedded Linux with kernel version 2.6.19. This operating system with ARM processor shows more deemed feasible performance in real-time embedded system than lower versions.

In the communication modules, major theme is to implement coexistence and multimodal communication among wireless LAN (WLAN), Bluetooth, and ZigBee RF communication. WLAN is used for network services such as internet, ftp, and e-mail, and Bluetooth is for wireless headset, mouse, and keyboard. The ZigBee is adapted for the location based service (LBS), which can be operated with ubiquitous sensor network systems. Because WLAN, Bluetooth, and ZigBee use the 2.4GHz ISM band, There are serious interference and coexistence problems among them. To minimize these problems we have designed and implemented the dynamic channel allocation algorithm in the device driver and application layers based on [6] and [7].

In the wearable computer, user interface should be easy to learn, easy to use, user-friendly, and environment-friendly. As the primary user interface of the UFC, our research group has developed ring-type gesture-based input device called iThrow. The appearance wearing the iThrow is shown in Fig. 2. The iThrow includes a three-axis accelerometer and a three-axis magneto-resistive sensor for recognizing a gesture and the direction of the finger [5]. It also has a ZigBee transceiver for transmitting the recognized gesture information to the UFC main module. By using the iThrow, we can intuitively and easily edit and upload a captured image.

III. UCC IN UFC

In this section, we present the design and implementation of an efficient and useful application for UCC in the UFC. The overall structure of our UCC in the UFC, called UCC-UFC, is shown in Fig. 3. There are four parts: Capture/preview, GUI interface, Communication, and iThrow interface. In the preview state, a sequence of small sized JPEG images inputs from the camera with a fixed frame rate. The capture/preview part uncompresses the JPEG images by a JPEG codec, and display them through a frame buffer interface. We implemented a GUI interface based on Tiny-X and a touch screen device.

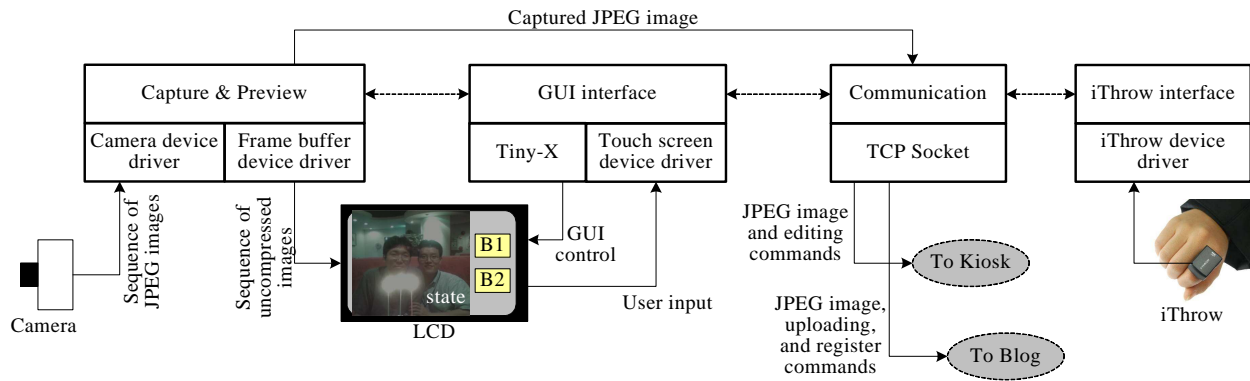


Fig. 3. Overall structure of UCC-UFC

By using this GUI interface user can control capture/preview and communication parts. Through TCP protocol, UFC can communicate with Kiosks and Blogs. We can send various editing commands to a Kiosk, and commands for automatic uploading and register images to blogs. Also, by using iThrow user can control our UCC-UFC comfortably and intuitively.

The detailed procedure of UCC-UFC is as follows.

- 1) *Capturing an image:* A user captures an image by using a camera module attached in the UFC. We used Logitech QuickCam Notebooks Pro[®] as the camera module because of its high quality and performance. Preview is displayed in the main module LCD as shown in Fig. 4. If User pushes "Pr/Ca" button through the touch-screen interface, current image is captured and stored as standard JPEG format, and preview is paused as the captured image. The user can make the preview play again by one more pushing the "Pr/Ca" button. The resolution of the preview is 320x240, but the image is captured as the resolution of 960x720.
- 2) *Sending the image to the blog or an kiosk:* If the user do not intend to editing the captured image, he can send the image to his blog by pushing "Blog" button. Otherwise, for editing the captured image, the user can send it to a near kiosk by taking throwing action to the kiosk, and then the image is displayed in the kiosk. The throwing action can be detected by the iThrow device. The kiosk is ready to edit the image. If the user pushes "Kios" button instead of taking throwing action, the image is sent to the nearest Kiosk. Since the editing job can be possible in the UFC with editing buttons as shown in Fig. 4, even though there is no kiosk near the user, he can edit the image in the UFC.
- 3) *Editing the image:* The user can edit the image; rotation or adjusting picture size, brightness, and contrast by taking various intuitive hand-gestures. We summarize and illustrate the possible hand-gestures and their image effects in Fig. 5. The gesture recognition algorithm is designed to be simple enough to run on a microcontroller inside the iThrow device by extracting the minimum set of required features and using threshold-based simple features. The delay yielded by the gesture recognition is too small users to feel [1] [5]. Note that all image effects in Fig. 5 can be

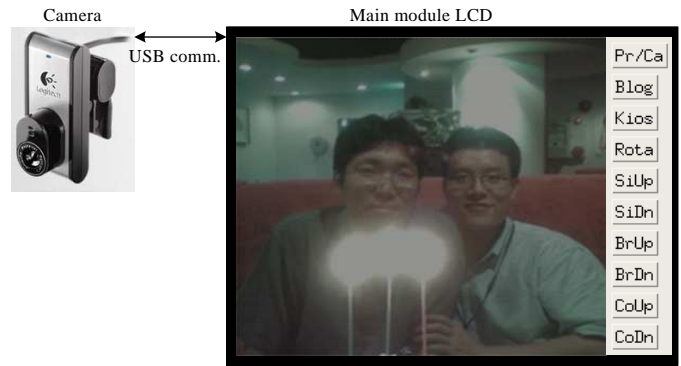


Fig. 4. Preview for capturing an image

applied by buttons shown in Fig. 4.

- 4) *Sending and registering the final image:* After editing, if the user takes the throwing action again, the final image is uploaded and registered in his blog automatically.

Generally, the blog site is managed by a major web portal company. Unfortunately, the process or APIs for uploading user's image is not opened to the public and difficult to be automated, because the uploading is intended to be performed only by human users. Nevertheless, to automate the uploading process, we need to snoop around the communication between user-side browser and the blog server, while letting a human user actually perform the uploading. First, we gathered cookies from a blog server. An usual blog server generally identifies its users with cookies, after the user logs in the blog server after typing his id and password. The user-side web browser maintains cookies to store any status or information to be preserved between the user and the server. Next, we collected the HTTP requests that the user-side web browser had sent; then we modified and replayed the requests to upload a new photo.

We used Logitech QuickCam Notebooks Pro[®] as the camera module because of its high quality and performance. The default output images of the camera are already compressed by standard JPEG format. For preview display, we must uncompress the camera output images in real-time. Even

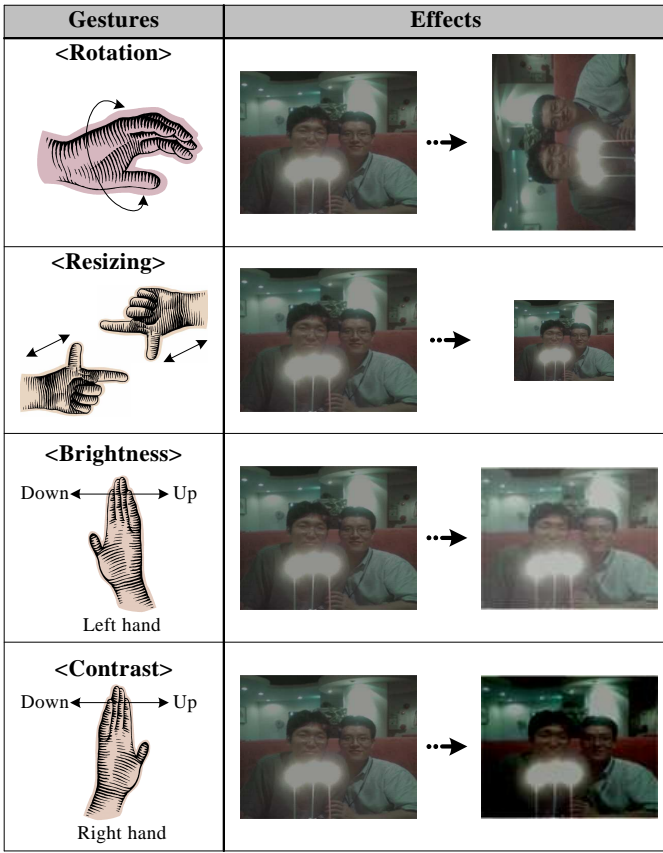


Fig. 5. The possible hand-gestures and their image effects

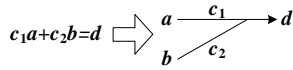


Fig. 6. Graphical notation for $c_1a + c_2b = d$

though the JPEG decoder we employ uses a well-known fast inverse discrete cosine transform (IDCT) for transforming the image data, because it was implemented by floating-point operations, the calculation overhead of the JPEG decoder is serious. Let's assume that the graphical notation as shown in Fig. 6 represents $c_1a + c_2b = d$. Also, if we assume that the input coefficients of i -th row and j -th column of an 8-by-8 block of IDCT are $b[i][j]$ and the output coefficients of i -th row and j -th column of an 8-by-8 block of IDCT are $b''[i][j]$, the fast IDCT algorithm used in the JPEG decoder is illustrated in Fig. 7 where $i, j = 0, 1, \dots, 7$. The multipliers c_k where $k = 0, 1, \dots, 7$ are predefined fixed floating-point values. There are two stages in the IDCT algorithm. In each stage, 8 times loops are performed by changing i and j from 0 to 7 respectively. Because in Fig. 7 all c_k where $k = 0, 1, \dots, 7$ are floating-point values, floating-point operations must be performed in the IDCT algorithm.

Because the core CPU of UFC has no floating-point operator, the calculation overhead of the IDCT is very heavy. For speed-up, we must change floating-point operations to

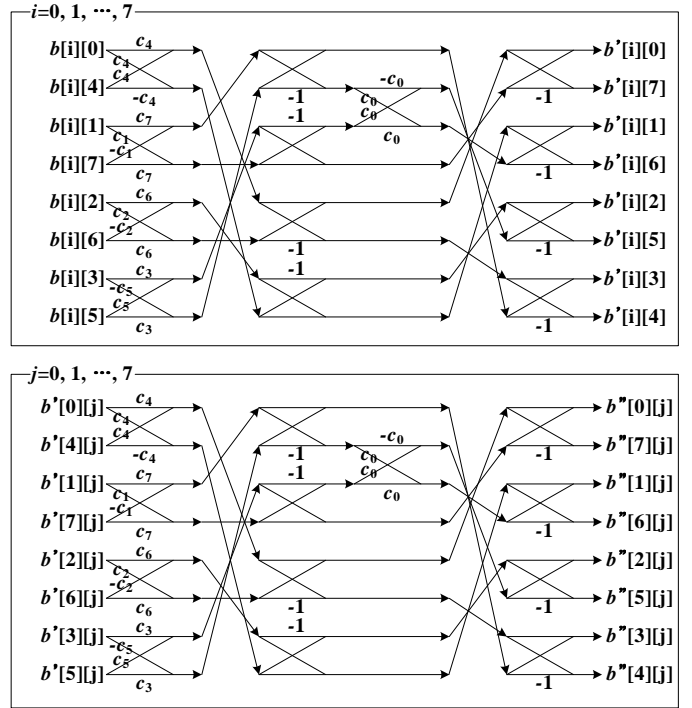


Fig. 7. The fast algorithm of the IDCT used in the JPEG decoder. $b[i][j]$ and $b''[i][j]$ are the input and output coefficients of i -th row and j -th column of an 8-by-8 block of IDCT respectively where $i, j = 0, 1, \dots, 7$. c_k where $k = 0, 1, \dots, 7$ are predefined fixed floating-point values.

integer operations. For that, we must scale up all $b[i][j]$ and c_k , and scale down all $b''[i][j]$ where $i, j, k = 0, 1, \dots, 7$. To reducing overhead, we performed scale-up operations by using left shift operations and scale-down operations by using right shift operations. The modified fast algorithm of the IDCT based on integer operations is shown in Fig. 8 in the case of the first stage. Note that c'_k is the integer part of $c_k \times 2^S$ where $k = 0, 1, \dots, 7$, and \ll and \gg are left and right shift operators respectively. The second stage of the IDCT algorithm can be modified by the same manner. By applying the modified algorithm, the processing speed increased more than 10 times, and the PSNR degradation was less than 0.1dB in our implemented system.

IV. CONCLUSION

In this paper, we presented the design and implementation of an application for efficiently creating and managing the user generated content in the UFC. First a user captures an image by using the camera attached in the UFC as shown in Fig. 1. For editing the captured image, the user can send it to a near kiosk by taking throwing action to the kiosk, and then the image is displayed in the kiosk. The user can edit the image such as rotation or adjusting picture size, brightness, and contrast by taking various intuitive hand-gestures. After editing, if the user takes the throwing action again, the final image is uploaded and registered in his blog automatically. Therefore, if users follow this scenario, they can fast and easily enjoy UCC services. Our application software is limited for

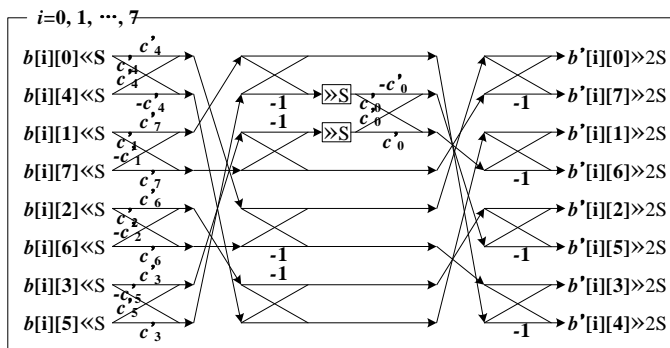


Fig. 8. The modified fast algorithm of the IDCT based on integer operations in the case of the first stage. c'_k is the integer part of $c_k \times 2^S$ where $k = 0, 1, \dots, 7$, and \ll and \gg are left and right shift operators respectively.

JPEG images. Further work is needed to expend to motion pictures such as MPEGx or H.26x.

REFERENCES

- [1] J. Lee, S. H. Lim, J. W. Yoo, K. W. Park, H. J. Choi, and K. H. park, "A Ubiquitous Fashionable Computer with an i-Throw Device on a Location-based Service Environment," *Proc. of AINAW*, May 2007, pp.59-65.
- [2] O. Amft, M. Lauffer, and S. Ossevoort, "Design of the QBIC wearable computing platform," *Proc. of ASAP*, Sep. 2004, pp.398-410.
- [3] Horizon Project, "User Created Content," <http://horizonproject.wikispaces.com/User-Created+Content>
- [4] K. H. Park, J. Lee, J. W. Yoo, S. H. Lim, and K. W. Park, "U-TOPIA: Campus-wide Advanced Ubiquitous Computing Environment," *Proc. of NGPC*, Jun. 2007.
- [5] J. W. Yoo, Y. W. Jeong, Y. Song, J. Lee, S. H. Lim, K. W. Park, and K. W. Park, "iThrow: A New Gesture-based Wearable Input Device with Target Selection Algorithm," *Proc. of ICMLC*, Jul. 2007.
- [6] IEEE802.15.2 Specification, "Coexistence of Wireless Personal Area Networks with Wireless Devices Operating in Unlicensed Frequency Bands," Aug. 2003.
- [7] M. Kang, J. Chong, H. Hyun, S. Kim, B. Jung, and D. Sung, "Adaptive Interference-Aware Multi-Channel Clustering Algorithm in a ZigBee Network in the presence of WLAN Interference," *Proc. of ISWCP*, Feb. 2007.