A Feasibility Study of Adopting Wireless Video for Wearable Computing

Ivan Lee[1], Zhenglin Wang[2], Bruce H. Thomas[3]
University of South Australia

ABSTRACT

Wireless technology can be adopted in wearable computing to facilitate many important applications, including the elimination of cabling and offloading process-intensive tasks to powerful servers remotely. However, a crucial challenge of adopting wireless technologies for wearable computing is the extra latency, which may be a critical issue for real-time mixed and augmented reality applications using optical or video see-through devices. This paper investigates the impact of latency while adopting video coding techniques to reduce wireless traffic. Trade-offs between processing complexity and latency for different augmented reality applications are investigated in this paper. Two video coding techniques using (1) standard H.264 video codec which yields superior compression efficiency, and (2) compressing sensing based video codec which demands low encoding complexity, are compared in this paper. The study investigates the feasibility of adopting low latency or low complexity design of future wearable devices facilitating optical or video see-through features.

KEYWORDS: Wireless networking, video coding, compressive sensing, mixed and augmented reality.

INDEX TERMS: H.5 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities.

1 INTRODUCTION

Facilitating augmented reality (AR) on portable devices such as smart-phones or PDAs has attracted attentions from industry and academia in the recent past. These readily available mobile devices eliminate the need for developing hardware prototypes, and hence shortening the time and effort for research and development. Research projects on top of smart mobile devices assume sufficient computing power are readily available, or will be available in the future, to perform complex operations.

The focus of this paper differs from the above-mentioned assumptions regarding the availability of abundant processing power. When AR applications perform beyond simple tracking tasks, and involve complex computer vision algorithms and access large database such as face recognition applications, the system may demand significant processing power and storage space which are impractical to be embedded on portable devices.

Remote computing was also proposed in a study of marker tracking for handheld augmented reality [1]. The approach proposed in this work is targeted for marker tracking, and thus the colour image is converted to black/white binary image and then compressed using run-length coding. Although the conversion and compression algorithm applied in [1] is low in complexity, the performance in terms of compression time and power consumption may not be very efficient by using general purpose processors which does not take advantage of vectorized data for the pixel-by-pixel colour conversion and run-length encoding. While advanced video codec are readily available on off-the-shelf smart-phone processors (either as DSP or ASIC), we argue that real-time video compression is achievable at similar level of power-consumptions.

It is debateable whether or not the mobile AR devices [2] should be equipped with high performance computing facility; similarly, the feasibility of using thin-client instead of PC, or netbook instead of high-performance laptops, is subject to personal preference. This paper assumes the need for a light-weight, ultra-portable, and inexpensive *wearable thin clients*. Thus, this paper investigates the feasibility of designing such portable devices which offloads heavy computational demand and excessive storage to a remote server. Specific scenarios under our investigation include latency sensible applications for video-see-through or optical-see-through wearable displays [3][4], as oppose to handheld devices which has more latency tolerance. Questions arising in this research include:

1. What are the most critical features for the portable wearable devices?
2. While it is mandatory to display a high quality video for photo-realistic immersive AR experiences, is it necessary to deliver the same video quality level for computers to process the data?

Low complexity and low latency are possibly the most popular answers to the first question. Therefore, we examine the properties of wireless video and its latency characteristics in each proposed scheme. For Question 2 above, we investigate the behaviour of adopting wireless video for marker based tracking applications. The experiment is done by generating encoded video with different quality levels for the remote server in each proposed scheme, and the remote server calculates the tracking information according to the coarse reconstruction and returns it to the portable wearable devices, then the portable wearable devices utilize tracking information to blend virtual objects onto the real scenarios captured by the wearable camera or goggles. In other words, the proposed system attempts to lower the latency by adopting video codec with high compression rate. The quality of the video can be degraded as long as the server can effectively detect the marker. The degraded video quality will be unnoticeable to the user, since the raw video will be directly displayed from either optical see-through or video see-through devices.

This paper compares the two video coding schemes using standard H.264 video codec and compressive sensing based video codec. H.264 is chosen for its popularity and ubiquity; compressive sensing is investigated because of the unique feature

[1, 3]: Ivan.Lee, Bruce.Thomas@unisa.edu.au
[2]: wanzy047@students.unisa.edu.au
Mawson Lakes, South Australia, 5095

20th International Conference on Artificial Reality and Telexistence (ICAT2010), 1-3 December 2010, Adelaide, Australia
ISBN: 978-4-904490-03-7 C3450 ©2010 VRSJ

of its reduced encoder complexity. The design of wearable computers using these two approaches can be (1) fast time-to-market using off-the-shelf H.264 codec, or (2) low power consumption devices by adopting compressive sensing. The objective of this paper is comparing the performance of the two coding technique, hardware implementation based on the study of this paper will be considered as a future work.

The rest of the paper is organised as follows. We firstly explain the design of the proposed system in Section 2. A proposed scheme based on H.264 is introduced in Section 3. Next, another proposed scheme based on compressive sensing is explored in Section 4. In Section 5, we examine the performance of two proposed schemes, followed by the conclusion.

2 Design of Wearable Thin Clients

Many of the existing wearable computing systems for AR applications ensure sufficient processing power to handle complex processing tasks. To meet this objective, excessive processing power, memory and storage spaces are integrated to the wearable devices, thus increasing the weight and making them less "wearable".

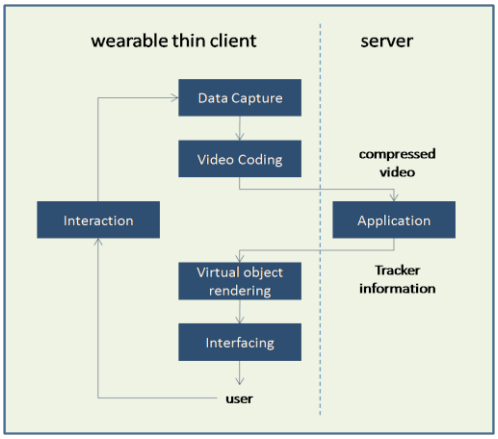


Figure 1. Processing stages for wearable thin clients

Another way to interpret wearable devices is that, these devices are designed to be ultra-portable. These devices possess just sufficient processing power to enable interfacing and interacting between computers and users. In other words, wearable computing could be considered as thin clients in the client/server architecture, with main functions to facilitate data capture (ambient information and user inputs), and data display (video, audio, haptics, etc).

The partition of the processing tasks on a client and on a server is illustrated in Figure 1. It is important to denote that since the user interaction (such as gesture or head movement) returns to the data capture block, latency minimization plays a major role for facilitating an immersive experience for virtual and augmented reality applications. Another point of interest is the virtual object rendering block. Other work in [1] suggests that this block could be handled on the server by moving it before the video decoding block. However, since 3D gaming are becoming popular on handheld devices, graphical processing unit is moving into the mobile processors just like video codec described before. Therefore, the power budget and processing time is improved with application specific integrated circuits (ASIC), and we believe the minimal feature set on a wearable thin client should be equip with sufficient 3D rendering capacities.

3 Wearable Computing Based on H.264

3.1 The Framework of Wearable Computing Based on H.264

Video is one major feature in today's smart phones, and advanced video codecs are readily available on modern mobile processors. Among these video codec standards, H.264 [5] (also known as MPEG-10 AVC) is prominent due to its high compression efficiency. A framework based on H.264 codec is easily set up according to the proposed system in Section 2. Figure 2 shows the block diagram based on the proposed H.264 scheme. A CCD or CMOS video camera is used to acquire the video samples. Then, the video samples are compressed using the H.264 codec ASIC and transmitted to the server via wireless transmission. The server reconstructs a degraded image with the received bitstream and computes tracking information. The tracking information is then reported to the wearable thin client. Another ASIC at the wearable thin client renders the virtual objects according to the received tracking information and displays the virtual object for the user. The H.264 video codec usually applies frame prediction technique to improve the compression efficiency. In general, mainly three types of frames (or their derived extensions) are used to encode videos: Intra (I) frames, Predictive (P) frames, and Bi-directional (B) predictive frames. The experiment conducted in Table 1 and Figure 3 applies H.264 codec, with different quality levels achieved by adjusting the quantization levels of these different types of frames (30 for high quality, 40 for medium quality, and 50 for low quality). The frame rate used for the experiment is 30

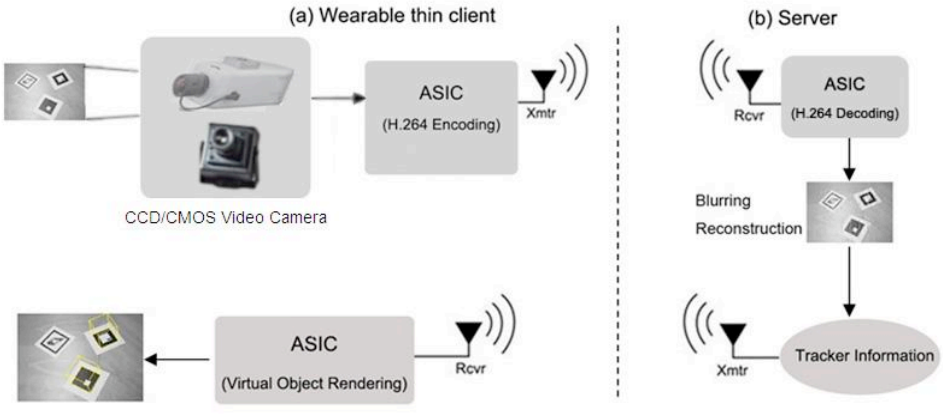


Figure 2. A proposed scheme based on H.264 for the wearable computing system

frames per second, and the bitstream data rate for different quality level and their corresponding quality measures using peak signal to noise ratio (PSNR) are illustrated in Table 1. Each pixel in the video frame is sampled using 8-bits, and PSNR measures the quality of the video respect to the peak value 255 (or 2^8) using the following formula

$$PSNR = 10 \log_{10} (255^2 / MSE).$$

The MSE is the mean square error between the original and reconstructed video frame. Figure 3 illustrates samples of the reconstructed videos in Table 1.

3.2 Latency Analysis

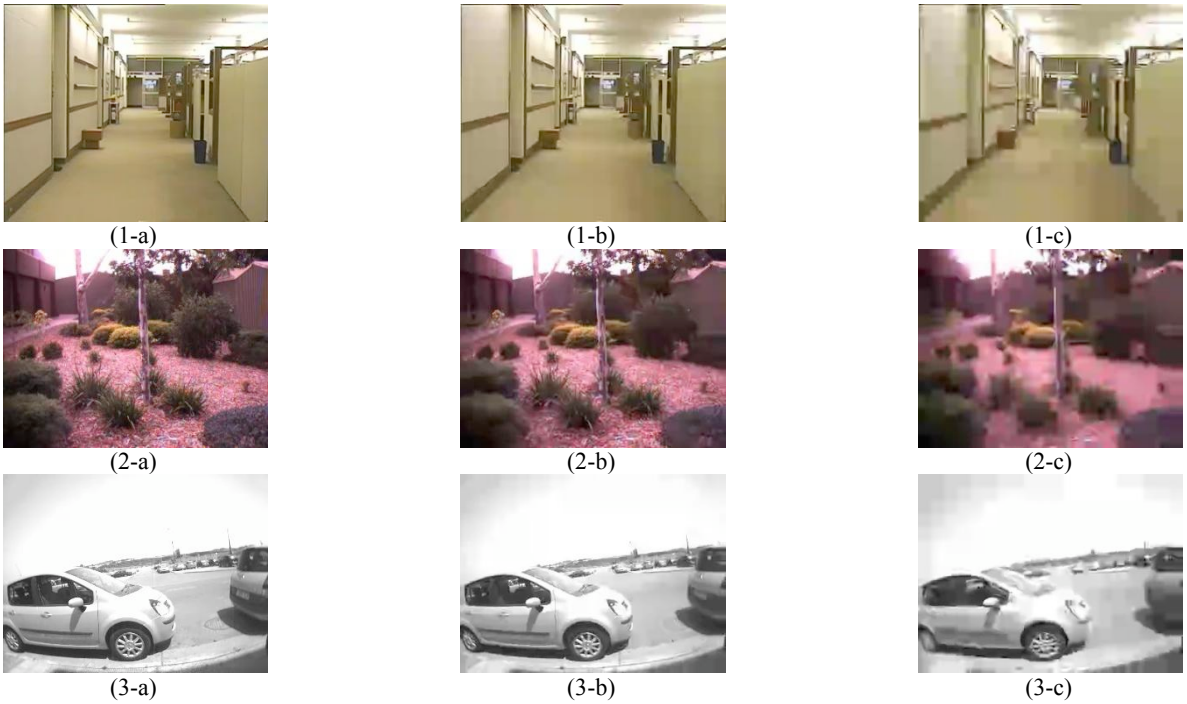
As discussed previously, low-latency or real-time is a critical requirement for wearable displays. For voice applications, one-way transmission should not exceed 150 msec, preferably below 100 msec for highly interactive applications, as recommended ITU-T's G.114 [6]. A sample latency of head mounted display studied in [7] has a mean value of 122 msec.

Candidate of wireless standards to be considered include IEEE 802.15.3 ultra wideband network, or IEEE 802.11n wireless local area network. The setup of the wireless communication should be using either the ad hoc mode with a point-to-point direct connection between the wireless node and the computing server; alternatively, if infrastructure mode is selected, it is desirable to connect the access point to the computing server on a LAN network. This is because both WLAN and UWB adopt collision avoidance techniques to share the physical transmission medium among different wireless devices. Eliminating wireless access point helps reducing collisions and scheduling, hence improving the performance.

Given that both 802.11n and 802.15.3 facilitate high data rate (802.11n at over 600 Mbps and 802.15.3 at up to 480 Mbps. Let r denote the utilization ratio of the entire transmission channel (due to protocol headers, contention-based and contention-free congestion avoidance protocols, and number of concurrent users, etc). Using the car sequence for instance, the transmission delay ranges from $0.1/r$ msec to $2.4/r$ msec. The maximum throughput ratio r is typically around 50%, depending on the protocol overhead and number of concurrent users. Assuming the channel is congested and r is 10%, the transmission delay will range from 0.17 msec to 3.5 msec (using the extreme values in Table 1). Note, the session setup time (such as protocol handshakes, authorization, authentication, and accounting) are ignored because these introduce an initial delay when establishing a session. The wearable thin client will not start rendering the virtual objects until the session setup is completed.

Table 1: Rate and quality of H.264 video compression

	Hall (CIF, 352x288)		Garden (RGB, 320x240)		Car (Monochrome, 320x240)	
	Bit Rate (kbps)	Quality PSNR Y (dB)	Bit Rate (kbps)	Quality PSNR Y (dB)	Bit Rate (kbps)	Quality PSNR Y (dB)
Raw	36495.36	N/A (infinite)	55296	N/A (infinite)	18432	N/A (infinite)
High Quality	193.05	38.14	1657.36	35.36	472.59	35.79
Mid Quality	31.21	30.94	207.76	26.89	59.94	28.94
Low Quality	9.99	24.31	51.01	22.00	19.84	23.39



(1-a) (1-b) (1-c)
(2-a) (2-b) (2-c)
(3-a) (3-b) (3-c)

Figure 3. Sample reconstructed video frames (1) hall sequence (2) garden sequence (3) car sequence. For each sequence, (a) represents the high-quality, (b) represents the medium quality, and (c) represents the low quality in Table 1.

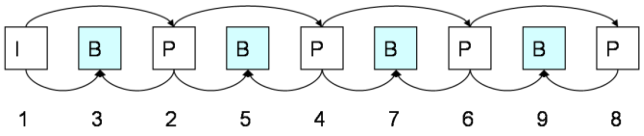


Figure 4. Video frame types and coding sequence

Assuming the hardware video coder encodes at real-time for video at 30 frames per second, the time to encode or delay each frame is at most 33 msec. The inter-dependencies between different frames for modern block-based video codec are illustrated in Figure 4, and the order of encoding sequence is indicated below each frame.

Thus, if multiple B frames are used, it is possible to introduce a framing delay. Let R denote the frame sampling rate, and let N_B denotes number of B frames between I-and-P or P-and-P frames, Assume the coder operates at real-time or faster. The coding delay (including framing) is

$$(N_B + 1)/R < T < 2(N_B+1)/R.$$

Because each B-frame introduces an additional 33ms delay, this accounts for a significant impact to the latency. Although B frame helps reducing the size of the compressed bitstream, the additional delay does not justify the low-latency requirement for augmented reality applications. Therefore, B-frames should be eliminated for minimizing the latencies, which is a crucial need for real-time AR applications. The extra latency over traditional head mounted display include the coding delay (less or equal to 33msec), the framing delay (33msec, and most of this latency overlaps with the existing latency), and transmission delay (less than 3.5 msec depending on the compression ratio and network utilization level). It is recommended to use a fast hardware codec which run beyond real-time to minimize the coding delay.

4 WEARABLE COMPUTING BASED ON COMPRESSIVE SENSING

4.1 The Framework of Wearable Computing Based on Compressive Sensing

Real-time compression based on H.264 can be achieved today using off-the-shelf processors, as commonly used on smart phones today. However, a common challenge for traditional transformation and motion-compensation based video codec such as H.264 is the high power consumption due to high-complexity encoding strategy. Recently, a new approach using compressive sensing has attracted great interest for video codec because of its low encoder complexity. Compressive sensing, also known as compressed sensing and compressive sampling, is an emerging technique for sampling and reconstructing a signal on the basis of the prior knowledge that the target signal is sparse or compressible [8][9][10]. The main idea behind compressed sensing is to directly exploit the sparse representations of interesting signals if they can be sparsely represented in another domain. The rising technique is introduced in this section to facilitate a wearable thin client.

Generally, a discrete image signal can be vectorized into a real-valued, finite-dimensional vector in Euclidean space. Provided that a column vector $X= [x_1, x_2, \cdots, x_n]^T$ denotes a discrete image signal. The literature of image signal processing has revealed that most natural images are sparse or compressible in the discrete cosine transform (DCT) domain [11]. Let Ψ denote the collection of DCT orthogonal basis, then,

$$X = \Psi S = \sum_{i=1}^{n} s_i \cdot \psi_i \quad (1)$$

where $S=[s_1, s_2, \ldots, s_n]^T$ is a sparse representation of X in the DCT domain and contains K non-zero DCT coefficients. $\Psi = (\psi_1|\psi_2|\ldots|\psi_n)$ is an n-by-n DCT transformation matrix with ψ_i being a column vector of Ψ.

The sampling process of compressive sensing is to use an m-by-n measurement matrix Φ to measure X and obtain an m-dimensional measured vector Y as (2). The components in the measured vector are called measurements. The measurement matrix $\Phi = (\varphi_1|\varphi_2|\ldots|\varphi_m)^T$ is mostly given to be a random matrix whose entries are established from Gaussian independent and identically distributed random variables of zero mean [9, 10] and φ_i denotes a row vector of Φ. Usually Θ is used to denote the product of Φ and Ψ.

$$Y = \Phi X = \Phi\Psi S = \Theta \cdot S \quad (2)$$

Generally, m is much smaller than n and comparable with K, which means the n-dimensional signal is reduced to an m-dimensional measured vector, so the sampling procedure embodies an inherent compressing process. If Φ and Ψ comply with the Restricted Isometry Property (RIP) and the number of measurements in the measured vector satisfies (3), the original signal can be ideally reconstructed with the measured vector Y [10].

$$m \geq O(cK\log(n/K)) \quad (3)$$

where c is a universal constant and K denotes the number of non-zero coefficients of the DCT representation.

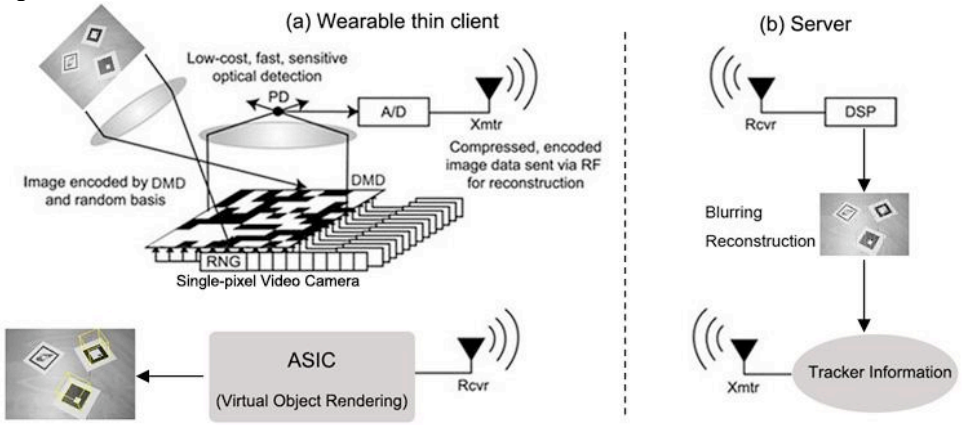


Figure 5. A proposed scheme based on compressive sensing for the wearable computing system (The single-pixel video camera component is adapted from http://www.dsp.ece.rice.edu/cscamera/)

Many advantages of compressive sensing are concluded in [12][13]. Among them, two merits might be advantageous to the wearable thin client:

i. Compression is built into the measurements. In other words, the samples acquired by compressive sensing can be transmitted to the decoder without compression process.
ii. Sampling is simple while reconstruction is complex. That means the heavy computation is shifted to the decoder so that the encoder maintains an advantage of low complexity.

The single-pixel camera developed at Rice University is an example of adopting compressive sensing for image compression [13]. Figure 5 shows the block diagram based on the compressive sensing scheme. A single-pixel video camera is equipped to acquire the original image. The acquired data are called measurements (also samples). Then, these measurements after A/D conversion are directly transmitted to the server via the wireless network. An optional entropy coding can be employed in the proposed scheme to further increase the compression ratio. In order to simplify the wearable client, the entropy coding strategy is unemployed in this proposed scheme. The server is assumed to have powerful computational capability. A degraded image is reconstructed at the server with the received CS measurements, but it is sufficient to identify the contours of the markers. The tracking information is consequently calculated and fed back to the wearable thin client. The ASIC at the client side renders the virtual objects base on the received tracking information and displays it to the user.

4.2 OMP with Sorted Random Matrix

Usually, the sampling process of compressive sensing is universal and low-complexity while the recovery algorithms are flexible but high-complexity. Many recovery algorithms have been proposed for the compressive sensing [14][15][16]. Among various recovery algorithms, orthogonal matching pursuit (OMP) is a popular choice due to its low computational cost and its ease of implementation [14]. It has been proposed to be applied in many image and video signal processing applications [17][18].

However, OMP usually needs more measurements than some other recovery algorithms in order to achieve the same reconstruction quality [19]. In other words, it is hard for OMP to achieve a high compression ratio. The basic idea of OMP is to pick the most correlated information with the target signal in a greedy fashion [14]. The most correlated information is generally corresponding to the largest DCT coefficient of the target signal. But sometimes OMP is unable to precisely locate which is the most correlated information due to some reasons mentioned in [20] so that it needs more measurements to revise these mistakes. As a common recovery algorithm of compressive sensing, OMP is suitable for any sparse signal recovery. The original signal is usually supposed unknown before it is reconstructed, including the positions and the values of non-zero coefficients. But, the sparse representations of most image signals are exceptional after they perform DCT transform. The literature of image signal processing reveals that the larger non-zero DCT coefficients are mostly located at the low-frequency positions [11] though the precise positions cannot be obtained prior to its recovery. Thus, on the basis of this common feature, a sorted random measurement matrix is proposed to assist OMP to locate the most correlated coefficient more precisely in [20].

The recovery algorithm of OMP with sorted random measurement matrix is introduced as follows. Let $S = [s_1, s_2, \ldots, s_n]^T$ be a sparse representation of the signal X in DCT transform domain. And there are K non-zero DCT coefficients in S. Let $\Theta = [g_1| g_2|\ldots| g_n]$, and g_i denotes the *i*-th column vector of Θ. And simultaneously the expansion of Θ is also expressed as follows:

$$\Theta = \begin{pmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{pmatrix} \quad (4)$$

Then $g_i = [A_{1i}, A_{2i}, \ldots, A_{mi}]^T$, and rewrite (2) as below:

$$Y = \Theta \cdot S = \begin{pmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{pmatrix} \cdot S = s_1 \cdot g_1 + s_2 \cdot g_2 + \ldots + s_n \cdot g_n \quad (5)$$

The basic idea of OMP is to find a vector g_i from Θ, in which direction Y can achieve the largest projection P. The vector g_i is named the most correlated vector with Y. Let Rf denotes the residual vector on the direction orthogonal to g_i, then,

$$Y = < Y, g_i > \cdot g_i + Rf \quad (6)$$

where $<Y, g_i>$ indicates the inner product of the vectors Y and g_i, which is the projection of Y on the direction of g_i if g_i is a unit vector. Thus, g_i is orthogonal to Rf. Suppose (g_1 , g_2, …, g_n) are all unit vectors, then,

$$\|Y\|^2 = |< Y, g_i >|^2 + \|Rf\|^2 \quad (7)$$

Note, the inner product of g_i and Rf is zero because they are orthogonal. Then, the largest projection P can be found by solving the following formula:

$$|P| = \max_i |< Y, g_i >| \quad and\ g_i \in \Theta, i \in (1,2,\cdots n) \quad (8)$$

The most correlated vector g_i is usually corresponding to the largest DCT coefficient s_i. Therefore, when the correlated vectors corresponding to the non-zero DCT coefficients are iteratively located, the sparse DCT representation of the original image signal can be retrieved with least-squares method [20].

The sorted random measurement matrix Φ is usually drawn from uniform distribution random function. Then, each row in this matrix is sorted in descending order. Next, this sorted random matrix can be applied to measure the target image signal. The sorted random matrix is universal so it can be repeatedly adopted for any image signal. Considered the sizes of most image signals are large, a block-based compressive sensing method [21] is employed in the experiment. The target image signal is divided into equal-size blocks, and each block is independently sampled and then reconstructed. At last, these reconstructed blocks are reorganized into a reconstructed image. This paper follows the same setup as in [21] by setting the block size as 32 in the experiment, and N is 1024. In order to simplify the experiment, we choose Ψ to be the DCT orthogonal basis. The sorted random matrix is universal and it can be repeatedly used for each image block. Meanwhile, the sorted random matrix is known by both the client and the server in advance, so it is unnecessary to transmit via the wireless network. The algorithm is summarized as follows:

Sampling:

1. Generate an M-by-N random matrix Φ whose entries are drawn from uniform distribution random function.
2. Sort each row of Φ in descending order.
3. Measure the target image signal with Φ: Y = ΦX

Reconstruction:

1. Initialize the residual Rf = Y, the matrix of chosen correlated vectors $\Omega = \{\}$, and the iteration counter t = 1

2. Find the index i of the most correlated vector via resolving (8). If the maximum occurs for multiple indices, break the tie because the iteration times might be beyond the number of non-zero coefficients in S or something else is disordered.

3. Build a new matrix of chosen correlated vectors $\Omega = \{\Omega, g_i\}$ and remove the chosen vector g_i from Θ.

4. Obtain a new approximation $\hat{S}$ via resolving the following equation with least-squares method.

$$Y = \Omega \cdot \hat{S}$$

5. The new residual is calculated as follows:

$$Rf = Y - \Omega \cdot \hat{S}$$

6. Increase t, go back to step 2 if $t<K$ or $\|Rf\|>\partial$

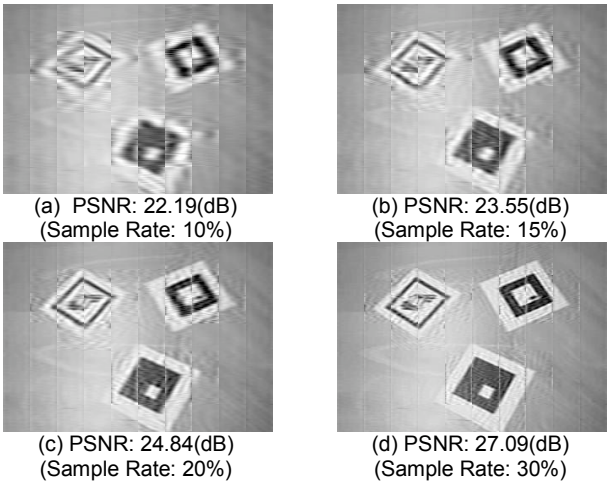
7. $\hat{S}$ is the approximate solution of S.

8. The original image signal can be retrieved with $\hat{S}$: $\hat{X} = \Psi\hat{S}$

Figure 6 illustrates some sample reconstructed video frames based on compressive sensing with different sample rates.

4.3 Latency Analysis

Because compressive sensing is an emerging technique, there are limited prior study on the latency of video codec and transmission based on compressive sensing. In this paper, comparisons between H.264 and compressive sensing based video codec for wearable computer are studied.



(a) PSNR: 22.19(dB) (Sample Rate: 10%)
(b) PSNR: 23.55(dB) (Sample Rate: 15%)
(c) PSNR: 24.84(dB) (Sample Rate: 20%)
(d) PSNR: 27.09(dB) (Sample Rate: 30%)

Figure 6. Sample reconstructed video frames based on compressive sensing with different sample rates

At the wearable thin client side, the wearable computing based on compressive sensing has the same structure as the wearable computing based on H.264 except that the CCD or CMOS video camera and the ASIC for H.264 encoding are replaced with the single-pixel video camera. The single-pixel video camera has been proved to possess the advantage of fast speed [13]. Therefore, the single-pixel video camera has the potential of yielding comparable or superior performance in terms of the latency than CCD or CMOS video cameras. On the other hand, the encoding latency is unavoidable for the wearable computing based on H.264. However, the encoding latency is inexistent in the wearable computing based on compressive sensing because the samples acquired by the single-pixel camera can be directly transmitted to the remote server. Hence, the wearable thin client based on compressive sensing possibly possesses the advantage of low latency than the one based on H.264.

At the server side, in order to increase the compression ratio, the P-frame and/or B-frame encoding strategy is usually adopted in the H.264 video codec scheme. The previous analysis reveals that if the B-frame encoding strategy is employed, each B-frame between I-and-P or P-and-P frames introduces an additional 33ms delay due to the order of I-frame, P-frame and B-frame. Furthermore, the latency of the reorder cannot be eliminated even if the server owns a powerful capacity of computation. In addition, if the P-frame encoding strategy is employed, the dependencies between P-frames and I-frames will increase the risk of the latency. If an error occurs in some I-frame, all the P-frames dependent on the I-frame have to be discarded. The server must be waiting until the next I-frame is received. In comparison, compressive sensing has an inherent advantage of high robustness. The measurements of compressive sensing are independent with one another. Losing a few measurements does not hurt the others and the reconstruction can still be carried out. Little effect is on the quality of reconstruction if the number of lost measurements is small [12]. If the computation ability of the server is sufficiently powerful, the latencies of decoding or reconstruction process can be negligible for both H.264 codec and compressive sensing. Consequently, the wearable computing based on compressive sensing is potential to exhibit a lower latency than the wearable computing based on H.264 with respect to the server.

In terms of the wireless network latency, the wearable computing based on H.264 presents a great advantage due to its high compression ratio. On the other hand, since the sample process of the wearable computing based on compressive sensing embodies an intrinsic compression, the compression process is usually suggested to be unemployed in order to maintain a low-complexity and low-power wearable thin client. But, when the sample rate of compressive sensing is about 10%, the encoded video based on compressive sensing just achieves the reconstruction quality equivalent to the low quality of the encoded video based on H.264 with compression ratio approximately 0.1%. Therefore, the wearable computing based on H.264 exhibits a lower latency than the wearable computing based on compressive sensing with respect to the wireless network.

However, the improvement of compressive sensing sample rate is promising along with extensive research. For example, a perceptual compressive sensing has been proposed to improve the reconstruction quality and/or sample rate in [22] while maintaining the characteristics of low-complexity and low-power encoder. In addition, if the bandwidth of the future wireless network is ultra-broad, the latencies of the wireless network will be negligible in comparison with the latencies spending on the wearable thin client and the server. Therefore, the wearable computing based on compressive sensing is potential to exhibit a lower latency than the wearable computing based on H.264.

5 CASE STUDIES

In this section, a popular AR application using marker-based tracking is examined by sending wireless video with different quality levels. A mathematical model is used as a substitution for single-pixel video camera. The raw video captured by the wearable camera is encoded using H.264 or compressive sensing. Then, the compressed video stream is transmitted to the server (a remote computer) for processing. Finally, the tracking information is generated at the server side and transmitted to the wearable thin client. It is possible that a low-quality video is transmitted to the

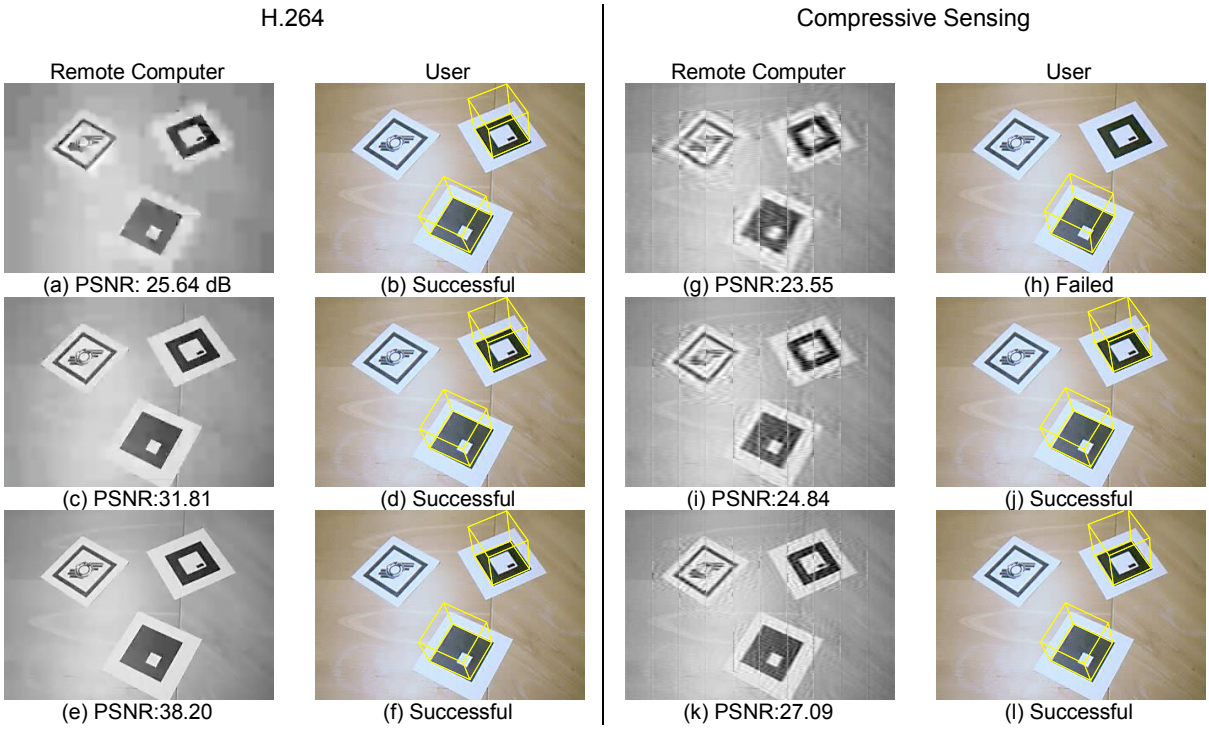


Figure 7. Marker tracking experiments

remote computer for the generation of tracking information, but the user can see the real scenarios blending with the virtual objects whose tracking information (virtual object type, orientation and position) is fetched from the remote computer. The real scenarios are captured by the wearable camera or goggles which are integrated as part of the thin client. In our experiments, all the markers are black and white so only the luminance data are necessarily sampled and transmitted to the server for the generation of tracking information. But, the user can see the colour scenarios blending with virtual objects. For the compressive sensing scheme, since the wearable goggles is still an open question, an additional conventional camera will be equipped to capture the real scenarios for blending with the virtual objects.

Figure 7 shows partial experimental results of marker-based tracking applications using the aforementioned two schemes at different quality levels: Sub-figure (a)-(f) show some sample video frames of the wearable computing based on H.264, and Sub-figure (g)-(l) show some sample video frames of the wearable computing based on compressive sensing; the monochrome pictures show the reconstructed video frames at the remote computer while the chromatic pictures show the video frames blending with the virtual cubes which are displayed to the user. The H.264 scheme exhibits an excellent performance in all test cases, and the target markers are detected and the orientation of the virtual cubes is properly found. However, some markers may not be tracked properly in the compressive sensing scheme when the sample rate is 15% and the video quality is 23.55dB in PSNR. It is hinted that a human intervention to manually control the video quality may be required.

6 Conclusion

In this paper we investigate the feasibility of adopting wireless video for wearable computing. We also investigate features of wearable thin clients which are equipped with limited processing capabilities, and applications are processed remotely. Two optional schemes are studied in this paper: the wearable computing based on H.264 and the wearable computing based on compressive sensing.

The wearable computing based on H.264 is not a fresh idea. The H.264 scheme exhibits high compression ratio and low network latency. However, the separation of sampling and compression will increase the complexity and power consumption of the wearable thin client. In addition, the frame prediction technique contributes the merit of high compression ratio, but introduces the demerits of high-complexity encoding and high latency of encoding and decoding process as well.

Video coding based on compressive sensing is proposed for wearable computers, since it demonstrates the advantages of low complexity and low power consumption in comparison with the H.264 scheme. The wearable computing based on compressive sensing consequently facilitates the wearable thin client more slender. On the other hand, the theoretical information in this paper reveals that low compression ratio of compressive sensing may introduce high network latency. So, the improvement of compression efficiency of compressive sensing is a potential research-worthy problem for the wearable computing based on compressive sensing. For example, a low-computation entropy coding technique can be imported to improve compression ratio.

For marker-based tracking applications, our observation suggests that it is possible to achieve identical results with video at low quality. Low reconstruction quality means high compression ratio so that the network latency is reduced. But the tracking performance may degrade if the quality is extremely low. Therefore, we suggest a manual control for video quality adjustment. Our study indicates that it is feasible to adopt proper quality video (with Y-PSNR values over 24dB) for real-time augmented reality applications, where low latency (framing, coding, and transmission delays) and high quality output (blending virtual objects onto the real scenarios captured by the

wearable camera or goggles) constraints are met for both the H.264 scheme and the compressive sensing scheme.

This paper focuses on the discussions of major design issues of wireless wearable thin client, and compares two different video coding approaches with different design objectives: (1) using readily available H.264 standard for fast time-to-market; and (2) using compressive sensing to reduce encoder complexity for wearable computers. Hardware implementation based on the study presented in this paper will be considered as potential future work.

REFERENCE

[1] D. Wagner and D. Schmalstieg. First steps towards handheld augmented reality. *Seventh IEEE International Symposium on Wearable Computers*, pp. 127-135, 2003.

[2] S. Woolley, J. Cross, S. Ro, R. Foster, G. Reynolds, C. Baber, H. Bristow, and A. Schwirtz. Forms of wearable computer. *IEE Eurowearable*, 2003.

[3] W. Pasman, S. Persa, and F. Jansen. Realistic low-latency mobile AR rendering. *Proceedings of the International Symposium on Virtual and Augmented Architecture*, Trinity College, Dublin, pp. 81-92, Jun. 2001.

[4] D. Perritaz, C. Salzmann, D. Gillet, O. Naef, J. Bapst, F. Barras, E. Mugellini, and O. Khaled. 6th Sense—Toward a Generic Framework for End-to-End Adaptive Wearable Augmented Reality. *Human Machine Interaction: Research Results of the MMI Program*, pp. 280-310, 2009.

[5] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H. 264/AVC video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13:560-576, 2003.

[6] ITU-T Recommendation G.114: One-way transmission time. *International Telecommunications Union*, 1996.

[7] R. Allison, L. Harris, M. Jenkin, U. Jasiobedzka, and J. Zacher. Tolerance of temporal delay in virtual environments. *IEEE Virtual Reality*, pp. 247-254, 2001.

[8] D. Donoho. Compressed Sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[9] E. Candès. Compressive sampling. *Proceedings of the International Congress of Mathematicians*, Madrid, Spain, 2006.

[10] E. Candes and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Transactions on Information Theory,* 52(12):5406–5425, 2006.

[11] T. Bose. Digital Signal and Image Processing. *John Wiley & Sons, Inc*, published, 2003.

[12] R. Baraniuk. Compressive sensing. *42nd Annual Conference on Information Sciences and Systems*, pp.4-5, 2008.

[13] M.F. Duarte, M.A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single-Pixel Imaging via Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):83-91, 2008.

[14] J. A. Tropp, and A. C. Gilbert. Signal Recovery from Random Measurements via Orthogonal Matching Pursuit. *IEEE Transactions on Information Theory*, 53(12):4655-4666, 2007.

[15] M.A.T. Figueiredo, R.D. Nowak and S.J. Wright. Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586-597, 2007.

[16] E. Cand`es and J. Romberg. ℓ1-magic : Recovery of Sparse Signals via Convex Programming. *California Institute of Technology*, Tech. Rep. 2005.

[17] V. Stankovi´c, L. Stankovi´c and S. Cheng. Compressive Video Sampling. *16th European Signal Processing Conference (EUSIPCO 2008),* Lausanne, Switzerland, August 25-29, 2008.

[18] D. Venkatraman and A. Makur. A Compressive Sensing Approach to Object-based Surveillance Video Coding. *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009.

[19] R. Chartrand and W. Yin. Iteratively Reweighted Algorithms for Compressive Sensing. *Acoustics, Speech and Signal Processing*, pp.3869-3872, 2008.

[20] Z. Wang and I. Lee. Sorted Random Matrix for Orthogonal Matching Pursuit. *Proc of International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2010.

[21] L. Gan. Block compressed sensing of natural images. *The 15th International Conference on Digital Signal Processing*, Cardiff, UK, pp.403–406, 2007.

[22] Yi Yang, Oscar C. Au, Lu Fang, Xing Wen and Weiran Tang. Perceptual Compressive Sensing for Image Signals. *IEEE International Conference on Multimedia and Expo*, pp.89-92, 2009.