2013

A MULTI-CAMERA MOTION CAPTURE SYSTEM FOR REMOTE HEALTH CARE MONITORING

Yun Ye
University of Nebraska-Lincoln

Song Ci
University of Nebraska-Lincoln

Aggelos K. Katsaggelos
Northwestern University

Yanwei Liu
Institute of Acoustics, Chinese Academy of Sciences, China

Multimedia and Expo (ICME), 2013 IEEE International Conference on
Year: 2013
Pages: 1 - 6, DOI: 10.1109/ICME.2013.6607566

A MULTI-CAMERA MOTION CAPTURE SYSTEM FOR REMOTE HEALTHCARE MONITORING

Yun Ye[1], Song Ci[1,2], Aggelos K. Katsaggelos[3], Yanwei Liu[2]

[1]Department of Computer and Electronics Engineering, University of Nebraska-Lincoln, USA
[2] Institute of Acoustics, Chinese Academy of Sciences, China
[3]Department of Electrical Engineering and Computer Science, Northwestern University, USA

ABSTRACT

This paper presents a multi-camera motion capture system aiming to provide caregivers with timely access to the patient's health status through mobile communication devices. The major components include video capture, object detection, video coding and transmission, error concealment, and video analysis. Our contribution is twofold. First, several novel ideas are developed, including fast object detection, and content-aware and adaptive video coding and transmission. Second, all components are seamlessly integrated in a unified optimization framework dedicated for online data transmission. In the scenario, the subject walked on a treadmill with four tripod cameras capturing the video from different viewpoints. After video compression and transmission over a wireless sensor network, the remote receiver recovered the videos and performed multi-view motion capture for gait analysis. Experimental results show that the presented system design achieves better video quality than traditional video coding and transmission scheme, while the requirement for a low-cost, noninvasive and real-time healthcare monitoring system is accommodated.

Index Terms— Healthcare monitoring, object detection, video coding and transmission, multi-view motion capture, wireless communications

1. INTRODUCTION

Remote healthcare monitoring is gaining increasing popularity due to the advances in multiple disciplines. One important task in a healthcare monitoring system is to provide a means to monitor walking patterns since it is a necessity for health evaluation of the neuromuscular system [1]. However, there are three major issues which prevent existing human gait monitoring systems from being used in the resource-limited environment such as rural clinics: 1) existing human motion capture systems using infrared sensing or other body sensing equipments are expensive. The average cost is around $250,000 which usually is not affordable for small clinics. 2) A motion capture system containing any body attachments, such as reflective or magnetic markers, gyroscopes and accelerometers, will be considered invasive, especially in geriatric attendance. 3) When there is interaction between the caregiver and the patient involved, e.g. instruction on how to adjust the gait, real-time transmission of the monitoring video is required. This issue is of primary concern when the communication resources are constrained.

Based on these considerations, we designed a marker-less motion capture system using multiple off-the-shelf cameras. This research is dedicated to developing a cost efficient remote healthcare monitoring system (through human gait analysis for neuro-health evaluation) at rural clinics in western Nebraska, based on our existing testbed of large-scale wireless multi-hop networks deployed in remote rural areas. The focus of this research is to study how to enhance the end-to-end video quality in an application-centric delay-constrained scenario through a cross-layer design method, by which video content analysis, video encoding/decoding, and video transmission are systematically considered. Therefore, multiple factors in the system level configuration are considered to determine the optimal video encoding and transmission parameters, including unequal error protection (UEP), transmission delay, quality balance, and error concealment.

The rest of the paper is organized as follows. Section 2 describes the system architecture and the formulation of the delay-constrained video coding and transmission problem. The fast object detection algorithm for UEP is introduced in Section 3. The content-aware video coding and transmission procedure is described in Section 4, and the adaptive video coding and transmission procedure is described in Section 5. The error concealment scheme by the receiver is explained in Section 6. In Section 7, the multi-view motion estimation process is described. Experimental results are provided in Section 8. Section 9 draws the conclusions.

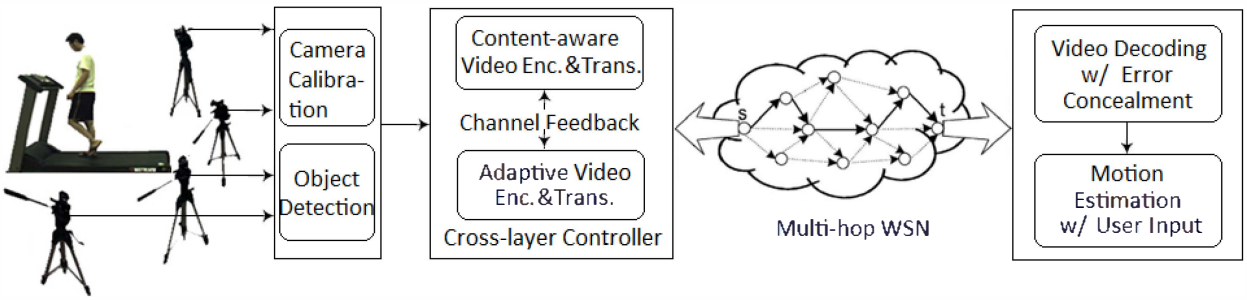


Fig. 1. Multi-camera motion capture system over WSN.

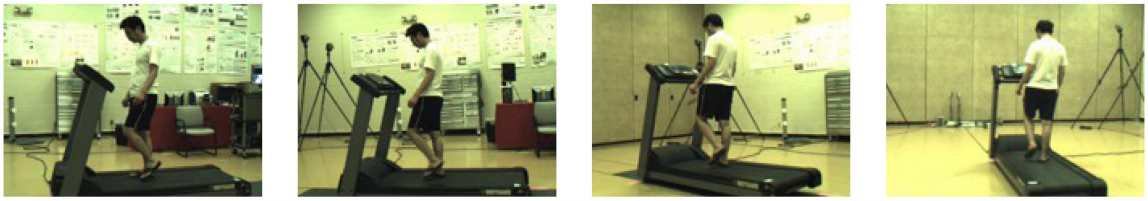


Fig. 2. Recorded video frame from four different views.

2. PROBLEM DESCRIPTION

The presented motion capture system for remote healthcare monitoring is illustrated in Figure 1. The videos showing the subject's walking pattern on a treadmill are recorded by four synchronized and calibrated tripod cameras from different viewpoints, as displayed in Figure 2. These videos are processed at the data center, i.e. the computer, where the ROI information is detected, and the parameters for video encoding and transmission are determined through cross-layer control. The multi-view motion estimation process is implemented by the receiver using the recovered videos and the camera calibration parameters [2]. To achieve optimal resource allocation, a content-aware video encoding and transmission procedure is applied by the cross-layer controller; and to ensure real-time video transmission, an adaptive encoding and transmission procedure is also applied concurrently based on the CSI. The number of cameras is limited for the consideration of cost and processing time. The cameras are sparsely positioned around the treadmill, and little inter-view correlation exists between different videos. Therefore, the four sequences of video packets are simulcast over the WSN.

At the cross-layer controller, the video encoding and transmission process is formulated as an end-to-end distortion minimization problem under a frame delay constraint:

$$\{s_{k,n}^*, c_{k,n}^*\} = arg\ min \sum_{k=1}^{K} \sum_{i=1}^{I} E[D_{k,n,i}]$$

$$s.t. \quad min \quad max_{k=1,2,\ldots,K}\left(\sum_{i=1}^{I} E[D_{k,n,i}]\right)$$

$$\sum_{k=1}^{K} \sum_{i=1}^{I} E\left[T_{k,n,i}(s_{k,n}, c_{k,n})\right] \le T^{max} \qquad (1)$$

Here $E[D]$ is the expected end-to-end distortion of one packet i, K is the number of views, and I is the number of packets in one frame. $\{s_{k,n}, c_{k,n}\}$ denotes the source coding parameter and channel transmission parameter vector for a frame n in view k. $E[T]$ represents the expected transmission time for one packet, and T^{max} is the maximum allowable delay for all the packets in one frame from K views to be transmitted.

Besides frame delay, another constraint is that, the maximum distortion of all the video frames should be minimized, i.e., the lowest quality is maximized, which also implicates a balanced quality among all the views. This constraint is necessary since the visual quality of each received video is considered to contribute equally to a successful 3D motion estimation process.

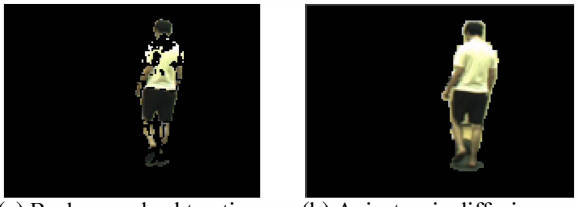
According to Formula (1), a best parameter vector $\{s_{k,n}^*, c_{k,n}^*\}$ is chosen for a new frame based on multiple factors affecting the expected distortion, including ROI, current channel condition, and previous packet loss information. Details of the solution procedures are explained in following sections.

3. FAST OBJECT DETECTION

Before video capture, the cameras are calibrated using the chessboard calibration pattern [3]. The calibration parameters are used for the 3D motion estimation at the receiver's side. After calibration, the object starts walking on the treadmill, and the motion videos are recorded by four synchronized cameras, and are analyzed to detect the object region in each view. A fast video object detection algorithm is implemented to bring out the ROI information, including background subtraction [4] and anisotropic diffusion [5].

3.1. Background subtraction

Background subtraction using Gaussian Mixture Model (GMM) is a popular video motion detection method known for its change adaptability and noise tolerance. GMM is an online learning process. Each pixel in a new frame is checked against the existing models until a match is found. A match is defined as the distance between the mean and the pixel value is within 2.5 times the standard deviation [4]. To accelerate the learning process, the background setting without moving objects is recorded at the beginning of the video, when sufficient data can be acquired to train the background models. Figure 3(a) shows the foreground detection results for one frame in one view.



(a) Background subtraction (b) Anisotropic diffusion

Fig. 3. Object detection.

3.2. Anisotropic diffusion

A problem with the temporal GMM based motion detection method is that it fails to detect some foreground regions with similar color to the background. As can be observed from Figure 2 and Figure 3, part of the body area is missing where the color of the T-shirt is close to the color of the wall. Spatial color correlation can be utilized to solve this problem, such as anisotropic diffusion [5]. Here anisotropic diffusion is applied as a post-processing step to improve the detection result. For example, a 4-nearest-neighbors discretization of the diffusion is expressed as

$$I_i^{t+1} = I_i^t + \lambda[c_N \nabla_N I_i^t + c_S \nabla_S I_i^t + c_E \nabla_E I_i^t + c_W \nabla_W I_i^t] \quad (2)$$

where I_i^t is the diffusion value at iteration t and at pixel i. λ is a constant between 0 and 1/4. N, S, E, W are subscripts for North, South, East, West. ∇I_i^t denotes the nearest-neighbor difference, and the conduction coefficient c is a kernel function of the Euclidean norm of ∇I_i^t,

$$c_i^t = f(\|\nabla I_i^t\|) \quad (3)$$

We design the kernel function as reversely increasing with ∇I_c, the color difference between adjacent pixels,

$$f(\|\nabla I_i^t\|) = \frac{w(\nabla I_C)}{\left(1+(A\|\nabla I_i^t\|)^2\right)\sum_{\aleph} w(\nabla I_C)} \quad (4)$$

$$w(\nabla I_c) = e^{-(B\|\nabla I_C\|)^2} \quad (5)$$

where A and B are predefined constants controlling the diffusion speed. $\aleph$ denotes the neighboring pixels. The diffusion value is initiated with GMM learning result, i.e., if a pixel i is detected as background, $I_i^0 = 0$; otherwise $I_i^0 = 1$. At the end of each iteration, resulting I_i^{t+1} is thresholded so that pixels with higher I_i^{t+1} value are determined as foreground. The iteration process is terminated either when the predefined maximum number of iteration is reached, or when the difference of the number of detected foreground between two successive iterations is below certain threshold, whichever comes first. Function (4) is a weighted version of the kernel function introduced in [5]. The merit is that if some region is missing, and it has neighboring foreground regions with similar color, its diffusion value will be raised continuously during the iterative diffusion process, making it more likely to be merged with those neighboring foreground regions. The final detection results are displayed in Figure 3(b).

The video object detection algorithm has an efficient implementation. For 300 recorded 640x480 frames from one view, the average processing time is 0.3 second per frame on a 32-bit PC machine with Intel E7300 2.66GHz CPU and 2GB RAM. The ROI region is defined as the smallest rectangle containing all the foreground pixels, aligning to the encoder block size. When the computation resource is constrained, only the data from one view is processed, the frames are down sampled (average processing time is 0.02 second per 160x120 frame), and the ROI regions for other views are projected using the camera parameters, and the input of the object's stature [2].

4. CONTENT-AWARE VIDEO CODING AND TRANSMISSION

The recorded videos endure data compression and transmission before arriving at the receiver. When the communication resources are limited in a WSN, an alternative of heavier compression is to implement unequal error protection (UEP) to impose higher priority on the parts of the video sequence that have a greater impact on video quality, e.g. the ROI [6, 7]. In the content-aware video coding and transmission procedure, the foreground data and the background data are grouped into different packets. While the sender applies the same compression and transmission parameters to all packets in one frame, the intermediate nodes in the WSN put a foreground packet ahead of all background packets in the queue. When a packet is lost, it will be retransmitted until it is correctly received, or discarded when the maximum transmission delay T^{max} is exceeded. As a result of the retransmission mechanism, the packet loss probability over a link between two nodes (u, v) mainly exhibits as the probability of packet drop due to delay deadline expiration when queuing at node u. Based on priority queuing analysis, it can be calculated from the tail distribution of the waiting time [8]:

$$p_{g,u} = Prob\left(E\left[W_{g,(u,v)}\right] + t_{g,u}^{0} > T^{max}\right) = \left(\sum_{g=0}^{1} \phi_{g,u} E\left[Z_{g,u}\right]\right) \cdot e^{-\frac{\left(T^{max} - t_{g,u}^{0}\right)\sum_{g=0}^{1} \phi_{g,u} E[Z_{g,u}]}{E\left[W_{g,(u,v)}\right]}} \quad (6)$$

$$g = \begin{cases} 0, & if\ it\ is\ a\ foreground \text{ packet} \\ 1, & if\ it\ is\ a\ background\ packet \end{cases} \quad (7)$$

where $t_{g,u}^{0}$ is the packet arrival time at node u, and $\phi_{g,u}$ is the average arrival rate of the Poisson input traffic into the queue at node u. $E[W_{g,(u,v)}]$ is the average packet waiting time at the queue of node u, and $E[Z_{g,u}]$ is the average service time at node u, measured as a geometric distribution with the effective transmission rate (goodput), packet length, and packet error and collision rate. Both the goodput and the packet error and collision rate are related to the link SINR (signal to interference and noise ratio) information and the selected modulation and channel coding scheme (MCS) [9]. Accordingly, the end-to-end packet loss rate (PLR) over a selected path *P* is estimated as

$$p_g = 1 - \prod_{(u,v)\in P}\left(1 - p_{g,u}\right) \quad (8)$$

The end-to-end packet delay is estimated as the sum of the packet delay $t_{g,(u,v)}$ over each link (u, v):

$$T_g = \sum_{(u,v)\in P} t_{g,(u,v)} = \sum_{(u,v)\in P}\left\{E\left[Z_{g,u}\right] + E\left[W_{g,(u,v)}\right]\right\} \quad (9)$$

The estimated packet loss rate and delay over each path are used by the cross-layer controller for optimal decision of coding and transmission parameters based on Formula (1). The solution strategy is summarized in next section.

5. ADAPTIVE VIDEO CODING AND TRANSMISSION

The multiple video sequences are simulcast over a multi-hop WSN. To accommodate the dynamic channel condition, flexible configuration of the video encoding and transmission parameters is enabled, including the selection of quantization parameter (QP), coding mode, MCS, and transmission path, resulting in a configuration quadruple (*Q*, *Mode*, *MCS*, *P*). In literature, how to choose the combination of the parameters for multiple sequences has been studied in various video streaming applications [10, 11]. Without the min-max (quality balance) constraint, the problem expressed in Formula (1) resembles the multiple-choice knapsack problem (MCKP) in classical combinatorial optimization [12]. In our application, the resource allocation is constrained by both transmission delay and quality balance. The expected video distortion is estimated with online CSI. And the optimal encoding and transmission parameters are configured by a cross-layer controller based on the distortion estimation results, using a greedy search algorithm.

5.1. Distortion estimation

When transmitted over the wireless network, the end-to-end distortion of a video packet includes the source coding distortion D^s and channel distortion D^c. Under a given configuration (*Q*, *Mode*, *MCS*), an optimal path *P* is selected based on the estimated video distortion, using the routing algorithm similar to the work in [9]. According to Equations (6) to (9), the estimated distortion for a packet π_g is

$$D_g(Q, Mode, MCS, P) = \begin{cases} E\left[\sum_{i\in\pi_g}\left(f_i - \tilde{f}_i\right)^2\right], & if\ \ T_g > T^{max} \\ D_g^s + D_g^c, & else \end{cases} \quad (10)$$

$$D_g^s = \left(1 - p_g\right) \cdot E\left[\sum_{i\in\pi_g}\left(f_i - \hat{f}_i\right)^2\right] \quad (11)$$

$$D_g^c = p_g \cdot E\left[\sum_{i\in\pi_g}\left(\hat{f}_i - \tilde{f}_i\right)^2\right] \quad (12)$$

f denotes the original data. $\hat{f}$ is the encoder recovered data after quantization. $\tilde{f}$ is the concealed data in the presence of packet loss. It is determined based on the receiver's packet loss feedback for previous frames. When the estimated packet delay is larger than the threshold, the concealment result is used to calculate the distortion directly. It is assume that perfect channel CSI is available to the sender without error and latency. This assumption could be approximately satisfied by using a fast feedback channel with powerful error control information as adopted in [13].

5.2. Parameter selection

From previous discussion, each configuration quadruple leads to a $\{D, T\}$ pair. It serves as an operation point for parameter selection. For each frame in a single view, the number of operation points is factored by the number of packets and available QPs, coding modes, and MCSs. To reduce the overhead, the packets in one frame share the same configuration. Maximum and minimum QPs for each view are tested under different coding modes and MCSs. The (*Mode*, *MCS*, *P*) configuration with minimum distortion is first selected for current frame in each view. To accommodate the video with the lowest quality, the selected (*MCS**, *P**) with maximum distortion among *K* views is assigned to other views. Then the maximum and minimum QPs are tested again under different coding modes and the assigned (*MCS**, *P**) to choose the optimal coding mode for each of the other views. After the (*Mode**, *MCS**, *P**) parameters are determined for each view, operation points using different QPs are generated, i.e. the number of operation points for each view is identical to the number of QPs, N_Q. The optimal QP is then chosen for each view according to Formula (1). To compare with the MCKP algorithm aiming at maximum sum product [12], the $\{D, T\}$ pair is transformed to $\{P, T\}$. *P* represents the quality (product), e.g. PSNR. It bears an increasing profile with *T* (weight). The solution procedure is listed in Figure 4.

1 Arrange the $\{P, T\}$ operation points $\{P_{jk,k}, T_{jk,k}\}$, $jk = 1, 2, \ldots, N_O$, $k = 1, 2, \ldots, K$, for each view in an increasing order. Remove the dominated points, i.e. $\{P_{jk,k}, T_{jk,k}\}$ is removed if $T_{jk,k} > T_{jk-1,k}$ and $P_{jk,k} \le P_{jk-1,k}$.
2 Select any one view k. Beginning with the point containing the highest weight that satisfies $T_{jk,k} < T^{max}$, perform the following greedy search:
(2.1) For each view h ($h \neq k$), find the point $\{P_{jh,h}, T_{jh,h}\}$, $P_{jh,h} \le P_{jk,k}$, $P_{jh+1,h} > P_{jk,k}$. If $P_{1,h} > P_{jk,k}$, $jh = 1$.
(2.2) Calculate the total delay $T_s = \sum_{iz \in \{j1,j2,\ldots,jK\}, z \in \{1,2,\ldots,K\}} T_{iz,z}$. If $T_s \le T^{max}$, go to step 2.4.
(2.3) If $jk = 1$, no solution exists. The program terminates. Otherwise set $jk = jk - 1$ and go to step 2.1.
(2.4) Sort the selected points $\{P_{iz,z}, T_{iz,z}\}$ according to increasing P. From the first one, calculate $T_{temp} = T_s - T_{iz,z} + T_{iz+1,z}$. If $T_{temp} < T^{max}$, set $T_s = T_{temp}$, replace $\{P_{iz,z}, T_{iz,z}\}$ with $\{P_{iz+1,z}, T_{iz+1,z}\}$, and repeat step 2.4. Else if $T_{temp} == T^{max}$, output all points, otherwise output current point.
(2.5) Output the selected combination $\{P_{iz,z}, T_{iz,z}\}$ and the corresponding QPs. The program terminates.

Fig. 4. Search for optimal combination of QPs.

6. ERROR CONCEALMENT

To counteract packet loss, error resilience and error concealment technologies are adopted to improve the video quality, including interleaving and boundary match. Before video encoding, interleaving is implemented to separate spatially neighboring MBs into different packets, as shown in Figure 5. For lost blocks in received video, the decoder performs boundary match [14] to search for similar patches in a spatiotemporal neighborhood. A patch yielding the smallest difference value in the search area is used to replace the missing MB, followed by a deblocking filter.

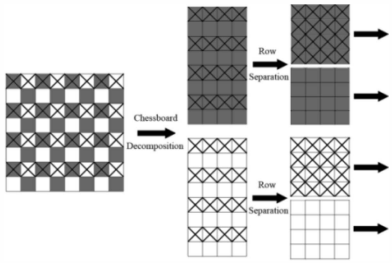


Fig. 5. Interleaving

7 MOTION ESTIMATION

The recovered video sequences are observed by the receiver. For motion estimation, the 3D positions of the object's joints are reconstructed using triangulation [15] based on the selected 2D coordinates from each view, as shown in Figure 6. Specifically, the projection from a point M in world coordinates (X, Y, Z) to a pixel (x, y) on an image plane is

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \backsim PM \quad \text{or} \quad \begin{cases} x = P(1)M/P(3)M \\ y = P(2)M/P(3)M \end{cases} \tag{13}$$

$P(i)$ is the i-th row of the camera projection matrix P. Equation (13) is equivalent to

$$\begin{bmatrix} P(3)x - P(1) \\ P(3)y - P(2) \end{bmatrix} M = AM = 0 \tag{14}$$

For K views, there is a system of equations according to Equation (14). The solution for M is obtained by singular value decomposition using the joint matrix $[A_1; A_2; \ldots; A_K]$.

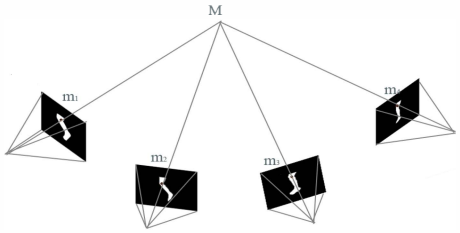


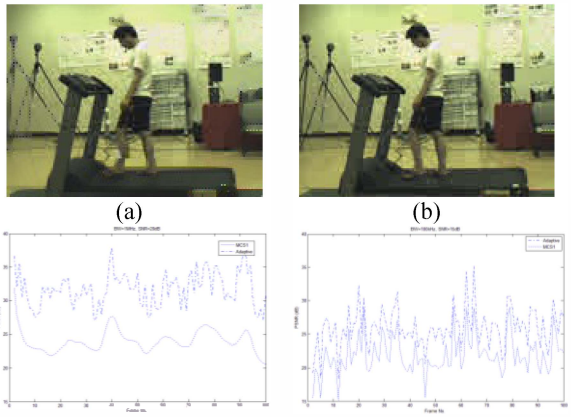
Fig. 6. Triangulation.

8. EXPERIMENTAL RESULTS

In our experiment, four tripod cameras (PointGrey Firefly MV) are placed around the object for video recording. The image size is 640x480. 100 frames from each view are processed. They are down-sampled to 160x120 to accelerate the computation. The video codec is based on the H.264/AVC standard [16]. The available QP set is {16, 20, 24, 28, 32, 34, 36, 38, 39, 40}. The MCSs include MCS1 (6, 2/3), MCS2 (4, 3/4), MCS3 (2, 1/2), and MCS4 (1, 1/2) with a packet size 1k bytes [17]. A 30-node network with a DAG-modeled connectivity structure and the Rayleigh fading channel [9] is simulated in MATLAB. The packet arrival rate at each node is set to 100 packets/s. To test the system performance under different conditions, the frame delay constraint is set to 15 fps and 30 fps, the average SINR is set to 15dB and 20dB, and the channel bandwidth BW is set to 100kHz and 1MHz.

The content-aware video coding and transmission procedure places higher priority on foreground packets. Under better channel condition (BW = 1MHz, $\overline{\text{SINR}}$= 20dB), the average PSNR for the ROI is 36dB under 15 fps delay constraint, and 32dB under 30 fps, 2-5 dB higher than the traditional coding and transmission scheme without priority. The adopted error concealment also has significant impact on the visual quality of the received videos. Figure 7(a) shows one recovered frame using the traditional scheme with slice copy as the error concealment measure. Compared to the result in Figure 7(b) obtained with the proposed method, the misplaced ankle could impose considerable error for the 3D motion estimation.

The adaptive coding and transmission procedure provides more accurate rate-distortion control under the dynamic channel condition, as demonstrated in Figure 7(c) and (d). The source coding scheme using a fixed MCS and



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

Fig. 7. Video coding and transmission.

view 1	view 2	view 3	view 4
12,1	9,1	11,2	10,3
15,2	13,3	12,4	13,5
16,3	15,4	14,5	14,6
17,4	16,6	18,6	17,8
19,5	20,7	21,7	20,10
22,7	23,8	24,9	23,11
25,8	24,9	28,10	26,13
26,10	28,11	29,12	32,14
32,18	35,20	38,20	36,23
39,27	40,31	45,22	40,29

Fig. 8. Operation points.

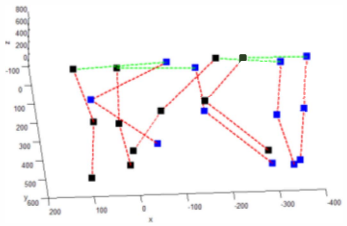


Fig. 9 Motion capture.

transmission path is compared with the proposed method, on the average PSNR of four views. The delay constraint is set to 30 fps.

The parameter selection procedure in Sec.5.2 achieves the min-max requirement as expressed in Formula (1). Figure 8 lists a set of {*P*(dB), *T*(ms)} operation points for one frame from four views. The total weight constraint is 30ms. The selected combination by the MCKP algorithm [12] is {15,2}, {15,4}, {28,10}, {32,14}. The result with our algorithm is {19,5}, {23,8}, {21,7}, {20,10}. The total product is lower, but the lowest quality is improved from 15 to 19, as well as the quality variance among different views.

Finally, to illustrate the motion capture process, the reconstructed 3D points at four different time instances are displayed in Figure 9. The blue markers represent the joints at the hip, knee, and ankle of the left leg, and the black markers represent the corresponding joints of the right leg.

9. CONCLUSIONS

The presented multi-camera motion capture system is designed for cost-effective, noninvasive and real-time remote healthcare applications such as gait analysis. Interdisciplinary study is conducted to incorporate different components of the system, including video object detection, data compression, wireless communications, and 3D reconstruction. Cross-layer control plays an important part in optimal system configuration, under the delay and quality requirements.

10. REFERENCES

[1] J.S. Rietman, K Postema, J.H. Geertzen, "Gait analysis in prosthetics (opinions, ideas and conclusions)," *Prosthet Orthot Int*., vol.26, pp.50–57, 2002.

[2] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *ICCV* 1999.

[3] J.Y. Bouguet, Camera calibration toolbox for Matlab, online available, http://www.vision.caltech.edu /bouguetj/calib_doc/.

[4] C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," *PAMI*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[5] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *PAMI*, vol. 12, pp. 629–639, 1990.

[6] H. Wang, F. Zhai, Y. Eisenberg, A.K. Katsaggelos, "Cost-distortion optimized unequal error protection for object-based video communications," *CSVT*, vol.15, no.12, pp. 1505- 1516, Dec. 2005.

[7] R. Chakravorty, S. Banerjee, and S. Ganguly, "MobiStream: error-resilient video streaming in wireless WANs using virtual channels," *INFOCOM* 2006.

[8] J. Abate, G. L. Choudhury, and W. Whitt, "Exponential approximations for tail probabilities in queues I: Waiting times," *Oper. Res.*, vol. 43, no. 5, pp. 885–901, 1995.

[9] D. Wu, S. Ci, H. Wang, and A.K. Katsaggelos, "Application-centric routing for video streaming over multi-hop wireless networks," *CSVT*, vol. 20, no. 12, pp. 1721-1734, Dec. 2010.

[10] J. Chen, A. Ghosh, J. Magutt, and M. Chiang, "QAVA: quota aware video adaptation", *ACM CoNEXT* 2012.

[11] L. Chen, J. Lehoezky, R. Rajkumar, D. Siewiorek, "On quality of service optimization with discrete QoS options," *RTAS* 1999.

[12] D. Pisinger, "A minimal algorithm for the multiple-choice knapsack problem," *European Journal of Operational Research*, vol. 83, pp. 394-410, 1995.

[13] Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems, *IEEE Standard 802.16*, 2002.

[14] B.W. Micallef, C.J. Debono, and R.A. Farrugia, "Performance of enhanced error concealment techniques in multi-view video coding systems," *IWSSIP,* pp.1-4, 16-18 June 2011.

[15] M. Pollefeys, L. Van Gool, A. Zisserman, A. Fitzgibbon (Eds.), 3D Structure from Images - SMILE 2000, *LNCS*, Vol. 2018, Springer-Verlag, 2001.

[16] H.264/AVC Reference Software, available online: http://iphome.hhi.de/suehring/tml/download/

[17] D. Krishnaswamy and M. van der Schaar, "Adaptive modulated scalable video transmission over wireless networks with a game theoretic approach," *MMSP*, pp.107-110, 2004.