Toward Long Distance Tabletop Hand-Document Telepresence

Chelhwon Kim
FX Palo Alto Laboratory
kim@fxpal.com

Patrick Chiu
FX Palo Alto Laboratory
chiu@fxpal.com

Joseph de la Pena
Fuji Xerox
joseph.delapena@fujixerox.com

Laurent Denoue
FX Palo Alto Laboratory
denoue@fxpal.com

Jun Shingu
Fuji Xerox
jun.shingu@fujixerox.co.jp

Yulius Tjahjadi
FX Palo Alto Laboratory
yulius@fxpal.com

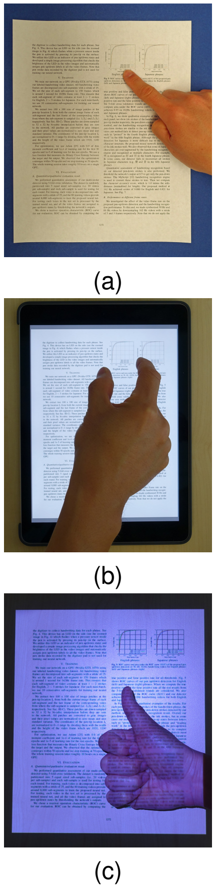


Figure 1: Hand gestures on a document: (a) a paper document, (b) a tablet, (c) a tabletop projection.

Abstract

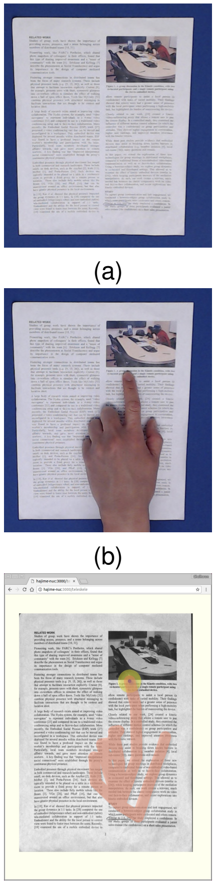
In a telepresence scenario with remote users discussing a document, it can be difficult to follow which parts are being discussed. One way to address this is by showing the user's hand position on the document, which also enables expressive gestural communication. An important practical problem is how to capture and transmit the hand movements efficiently with high resolution document images. We propose a tabletop system with two channels that integrates document capture with a 4K video camera and hand tracking with a webcam, in which the document image and hand skeleton data are transmitted at different rates and handled by a lightweight Web browser client at remote sites. To enhance the rendering, we employ velocity based smoothing and ephemeral motion traces. We tested our prototype over long distances from USA to Japan and to Italy, and report on latency and jitter performance. Our system achieves relatively low latency over a long distance in comparison with a tele-immersive system that transmits mesh data over much shorter distances.

Author Keywords

Telepresence; hand gestures; document sharing.

Introduction

For a media rich teleconference, multiple streams for the video, audio, document, and other contents can be incor-



(a)

(b)

(c)

Figure 2: Options for viewing remote hand feedback over paper document page: (a) no feedback, (b) video of hand interacting with document, (c) detected skeleton transmitted and rendered on hi-res captured document image.

porated. A typical scenario for discussing a document may use either an audio or video channel along with a live display of the document. The document may be in the physical form of paper, or in digital form viewed on a tablet computer or projected on a tabletop surface (Fig. 1).

The document conferencing experience can be improved by showing the hand over the document. In terms of communication purposes, capturing and displaying the movements and gestures helps the users to disambiguate and interpret speech and actions; and in terms of pointing and deictic gestures, being able to directly interact with the object is better than using a separate device like a mouse which inhibits users from making these useful gestures [4].

One way to support hand gestures over a document is to allow the user to point at parts of the document, and use a videoconferencing system to show this at the remote site (see Fig. 2 b). However, for paper documents, existing standard videoconferencing technology do not support sufficiently high resolution to read the document page. Thus, a challenging problem is how to transmit hand movements efficiently along with high resolution document images.

In this paper, we propose a system with two channels that integrates document capture with a 4K video camera and hand tracking with a webcam, in which the document image and hand skeleton data are transmitted at different rates and handled by a lightweight Web browser client at the remote sites.

The hand skeleton is captured with a state-of-the-art 3D hand pose estimation algorithm based on deep neural network [8] using a commonly available low-cost webcam. The skeleton data comprises a small set of numbers describing a few line segments and can be transmitted at a high frame rate without requiring a video codec. In conjunction with the skeleton information, the document image is captured with a high-resolution 4K camera (e.g. [3]) and transmitted only when the document page has changed.

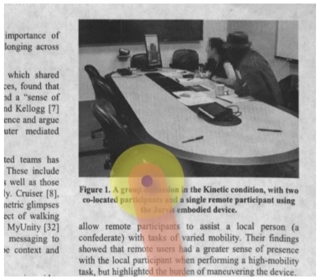
At a remote site, both channels are handled by a JavaScript client in a Web browser (see Fig. 2 c). A close-up of the captured hi-res document with the rendered hand skeleton is shown in Fig. 3. To minimize occlusion, we use translucency in the rendering. In addition, to make it easier to follow the hand motions, we employ velocity based smoothing and ephemeral motion traces.

We tested our prototype over long distances from USA to Japan and to Italy, about 5100 miles and 6000 miles respectively. We collected data from 15 test sessions for each site, and report on latency and jitter performance. Due to our small skeleton data size, we can achieve relatively low latency in comparison with a tele-immersive system [5] that transmits mesh data over much shorter distances.

Related Work

KinectArms [2] is a toolkit for capturing and displaying arm embodiments for distributed groupware. It uses a depth camera to segment the video and sense the gesture height. It provides a variety of visual representations with arm skeleton structure data, and renders the arms with translucency level mapped to arm height. The toolkit was not evaluated in a setup with a remote site.

ShadowHands [6] uses a Kinect depth camera and an iterative model-fitting optimization algorithm to capture gestures, which are visualized remotely with a high-fidelity hand model. Hand rendering styles include a cartoon-like hand and a realistic bone-like skeleton hand. A study was done with two connected sites located in the same room.



importance of
longing across

which shared
ces, found that
nd a “sense of
nd Kellogg [7]
ence and argue
uter mediated

ted teams has
These include
s well as those
ly. Cruiser [8],
netric glimpses
ect of walking
MyUnity [32]
messaging to
e context and

Figure 1. A group discussion in the Kinetic condition, with two co-located participants and a single remote participant using the Jarvis embodied device.

allow remote participants to assist a local person (a confederate) with tasks of varied mobility. Their findings showed that remote users had a greater sense of presence with the local participant when performing a high-mobility task, but highlighted the burden of maneuvering the device.

Figure 3: Close-up of the captured hi-res document with the rendered skeleton. We use translucency to minimize occlusion.

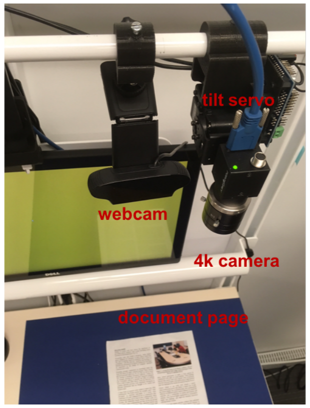


Figure 4: Our system uses a 4k video camera mounted on a tilt servo for capturing hi-res document on a tabletop. A low-cost webcam is used for tracking document page and user’s hand.

H-TIME [5] is a tele-medicine system for a doctor to examine a patient issues with the arms and shoulder. It uses a Kinect depth camera to capture 3D mesh data, and a haptic device to provide force feedback. A study was done between two sites that were 27 miles apart. The mesh data latency was in the range of 50ms to 400ms, and the haptic data latency averaged 3.7ms with large variations and with some jitters over 100ms.

These systems (KinectArms, ShadowHands, H-TIme) all require special depth cameras, wheres our system uses a commonly available low-cost webcam. For actual remote testing, only H-Time connected over remote sites; in comparison, our system is tested over much longer distances (2 orders of magnitude farther).

Hand-Document telepresence system

The hardware setup is shown in Fig. 4 and an overview of the proposed system is shown in Fig. 5. Our system has two channels that integrates document capture with a 4K video camera and hand skeleton & document page tracking with a webcam, in which the document image and hand skeleton data are transmitted at different rates. For the hi-res document page capturing part (Left panel in Fig. 5), we installed a 4k video camera mounted on a servo above the desk top similar to the system proposed by [3]. See Fig. 4. The system automatically steers the camera on the servo to capture a sequence of hi-res document page images, and the captured images are stitched together using a keypoint matching based image fusion algorithm as in [3]. The stitched image is further processed to correct the perspective distortion to get a fronto-parallel document page image. See the rectified image in Fig. 5 left panel. The high resolution document image is then stored in a server and is streamed from the server to the remote user’s web browser.

After the hi-res document image is streamed to the remote site, our system uses a webcam installed above the desk (next to the 4k camera as in Fig. 4) to capture the user’s hand interacting with the document page. We use the state-of-the-art deep learning based hand pose estimation method by [8] to obtain the hand skeleton, and use a rectangle detection method similar to [7] to detect the document page’s boundary. See the color encoded hand skeleton overlaid on the user’s hand and the red quadrilateral in Fig. 5 right panel. In order to get temporally smooth hand tracking motions, we filter the output of the hand pose estimation with the 1-Euro filter [1].

The hand skeleton coordinates are transformed to the rectified hi-res document page image coordinate system by applying a projective transformation (homography) between the quadrilateral of the detected document page and a canonical square box (Fig. 5 Right). At the remote site, the transmitted hi-res document page image is rendered on a web browser’s HTML canvas, and based on its dimension and location on the web browser ((W, H) and v in Fig. 5 Bottom respectively), the hand skeleton data is rescaled and translated.

Since the data size of the hand skeleton data message is about 2.6kB (21 hand skeleton joint coordinates, each of which is a float data type 2-vector, and other attributes such as color and timestamp), it can be transmitted at a relatively high frame rate; whereas the hi-res document page image (∼12MB) is streamed to the client only when the page has changed.

A lightweight client app running in a Web browser at the remote site renders the hand skeleton translucently over the hi-res document page image (Fig. 5 bottom).

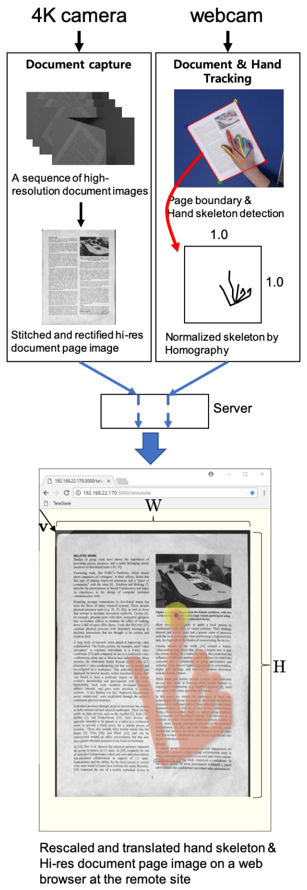

4K camera
webcam
Document capture
Document & Hand Tracking
A sequence of high-resolution document images
Page boundary & Hand skeleton detection
1.0
1.0
Stitched and rectified hi-res document page image
Normalized skeleton by Homography
Server
192.168.22.170:3000/teleskele
TeleSkele
W
V
H
Rescaled and translated hand skeleton & Hi-res document page image on a web browser at the remote site

Figure 5: System overview.

Data Transmission
The captured document images and the hand skeleton data are transmitted separately to the Web browser client.

The document images are captured by the system and stored on a server. These high resolution document images have sizes ranging in megabytes. Prior to data streaming, the client's web app is notified via an event message that a document is available for viewing. Once the web app acknowledges the event message, it can use the URL (enclosed in the event) to start the data streaming.

The skeleton data is streamed as a series of rapid transmission events. The skeleton data is encoded in JSON format. A typical skeleton data message is just a few kilobytes, making it possible to transmit at a higher frame rate using the Socket.IO real-time communication engine.

Hand Skeleton Data Rendering
The HTML Canvas element is used as a container for graphic objects such as images and lines. Each object has its own drawing properties such as length, width, position and color.

Given the flexibility of HTML using Cascading Style Sheets, the system stacks two canvas layers. The bottom canvas shows the document image. The top canvas shows a rendering of the hand skeleton. Separating the two objects in different canvases makes it possible to have a steady document background and only render the skeleton data without having to clear and redraw the document image every time new skeleton data is received.

JavaScript is used to do the drawing of the document image and the translucent lines of the hand skeleton on the canvas elements. The location of the index finger is highlighted with a translucent yellow filled circle. When new hand skeleton data is received from the server, the script clears the top canvas and redraws the entire hand skeleton with different coordinates.

An ephemeral motion trace is also drawn in the top canvas, showing the past positions of the index finger. The past positions are rendered with gradient effect to show a smoother animation.

Evaluation

The system was tested in two distributed environment setups over long distances: Palo Alto/US $\sim$ Yokohama/Japan (approximately 5100 miles), and Palo Alto/US $\sim$ Verona/Italy (approximately 6000 miles).

System Setup
For each setup, a local user in Palo Alto runs the system's server and the remote user connects to the server via the web browser. The computer in Palo Alto is a Windows PC, the client in Yokohama is a Windows laptop, and the client in Verona is an Apple laptop.

The local user starts the document capture app to capture and reconstruct the high-res document page image with a 4k camera. The document page image is sent to the remote user via the server and rendered on the remote user's web browser. The local user then starts the hand-document tracker app to track the hand and the document page with a webcam. The app runs on a desktop Intel Core i7-7700k CPU@4.20GHz, 16GB RAM, NVIDIA GTX 1070, and sends the hand skeleton data to the server at a rate around 6 fps. The tracked hand skeleton data is transmitted to the remote user and is rendered on the remote user's web browser over the hi-res document page image.

For each setup, we conducted 3 test sessions on 3 different days but at the same time of day during work hours (Palo Alto: 5:00pm, Yokohama: 9:00am; Palo Alto: 10am, Verona:

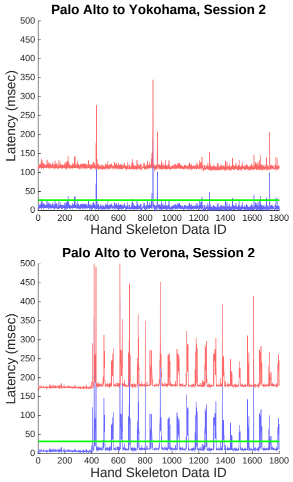


Figure 6: The upper (red) and lower (blue) bounds of the one-way system latency. Top: From Palo Alto/US to Yokohama/Japan, Bottom: From Palo Alto/US to Verona/Italy. Green line: latency of the light.

7:00pm). The Yokohama and Verona sessions were done on different days. Each session had 5 tests, and for each test, we collected the hand skeleton data for 1 minute.

Estimating Latency

During the data collection, the system logs a *broadcast* timestamp in the local user's time zone (i.e. PST time) when the hand skeleton data sample is sent to the remote user. When the client app running on the remote user's web browser receives the hand skeleton data, it creates and sends back a message with a *client* timestamp in the remote user's time zone (i.e. Japan or Italy time). The system in the local site then logs the received client timestamp message. Finally, the system logs a separate *acknowledgment* timestamp in the local user's time zone right after receiving the client timestamp.

The computers at each site are synchronized using their operating systems' time service to nearby network time servers; however, over such long distances the times may not be perfectly aligned (after adjusting for the time zones). Note that synchronizing to a single specific NTP time server is also problematic because of the long distances involved – it takes about 30ms for speed of light to travel that far. To deal with this phenomenon, we first find upper and lower bounds of the time difference between two distributed sites by analyzing the collected data and then use those bounds to estimate the latency.

Let us denote the broadcast timestamp as b_i, the client timestamp as c_i, and the acknowledgment timestamp as a_i for i-th hand skeleton data. The conversion of the client timestamp (from the remote to the local) can be done by adding the time difference d as $c_i + d$. From the ordered sequence of events, we look for d satisfying the following inequality for all hand skeleton data samples: $b_i < c_i + d < a_i$ for all i, and $b_i - c_i < d < a_i - c_i$ for all i. Hence, the upper and lower bounds of d can be obtained by finding the maximum/minimum of the bounds in the above inequality for all i: $\max_i \{b_i - c_i\} < d < \min_i \{a_i - c_i\}$

Results

The one-way system latency $l_i = (c_i + d) - b_i$, and with the computed lower and upper bound of d, of all samples for one test session, is shown in Fig. 6. Over such long distances, the speed of light latency is noticeable (green line in figure).

Table 1 shows the upper and lower bounds of the mean (along with the standard deviations) of the system latency for all test sessions. The system's jitter in the table is defined as the delay between two consecutive hand skeleton data samples at the receiving side. While the data transmission latency between US and Japan shows some large variations (especially for Session1 and Session2), the latency between US to Italy shows more stable (i.e. small standard deviations). The average of the means of the upper bound (worst latency) across all test sessions in Table 1 is 155 msec from Palo Alto/US to Yokohama/Japan (approximately 5100 miles), and 189 msec to Verona/Italy (approximately 6000 miles). The network latency by the speed of light from Palo Alto to Yokohama is around 27 msec, and to Verona is around 32 msec (green line in Fig. 6).

Table 2 shows the high-res document page image streaming latency to the remote site. The average latency is 20.9 sec from Palo Alto to Yokohama, and 22.4 sec from Palo Alto to Verona.

The average system latency for transmitting the hand skeleton data and the hi-res document page data from Palo Alto to Yokohama is smaller than the ones from Palo Alto to Verona, which is consistent with the distances of two remote sites from the local site.

Table 1: Data transmission statistics of hand skeleton data from US to Japan and to Italy for three sessions (S1,S2,S3). For Latency, the first two numbers are the lower and upper bounds of the mean. For Jitter, the first number is the mean. The numbers in parenthesis are the standard deviations.

Palo Alto-Yokohama

	Latency	Jitter
S1	78∼180 (111)	79 (77)
S2	9∼115 (11)	6 (10)
S3	74∼171 (82)	78 (66)

Palo Alto-Verona

	Latency	Jitter
S1	11∼189 (21)	8 (19)
S2	20∼188 (31)	9 (26)
S3	23∼191 (46)	13(41)

(msec)

Table 2: Data transmission latency of document page image (12M) from US to Japan and to Italy.

Palo Alto-Yokohama

	Latency
S1	24.44
S2	11.71
S3	26.53

Palo Alto-Verona

	Latency
S1	24.68
S2	13.96
S3	28.48

(sec)

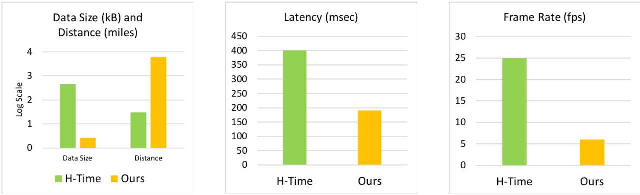


Figure 7: Comparison of our system with H-TIME [5].

We compare our system with the H-TIME [5] tele-immersive system in terms of transmission data size, distance between two distributed sites, one-way latency, and data frequency (see Fig. 7). Due to our small skeleton data size (∼2.6kB, 2 orders of magnitude smaller), we can achieve relatively low latency (∼190 msec) over much longer distance (∼6000 miles, 2 orders of magnitude farther) compared with H-time (∼400 msec) that transmits mesh data (∼450 kB) over much shorter distances (∼30 miles). The frequency of data transmission is 6 fps in our system, and 25 fps in H-TIME.

Conclusion and Future Work

We presented a novel system for hand-document telepresence with high resolution document capture and hand skeleton tracking, and with two separate channels for transmitting these data. We evaluated our system over long distances, and compared our system to a tele-immersive system that was tested over much shorter distances.

For future work, one direction is to evaluate our system on user level tasks over long distances, and to use our system in realistic situations with remote colleagues. Further enhancements to our system include optimizing the hand and document trackers to improve the frame rate, and supporting multiple hands over a document from multiple sites.

REFERENCES

1. Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1-Euro filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proc. CHI 2012*. 2527–2530.
2. Aaron M Genest, Carl Gutwin, Anthony Tang, Michael Kalyn, and Zenja Ivkovic. KinectArms: a toolkit for capturing and displaying arm embodiments in distributed tabletop groupware. In *Proc. CSCW 2013*. 157–166.
3. Chelhwon Kim, Patrick Chiu, and Henry Tang. High-Quality Capture of Documents on a Cluttered Tabletop with a 4K Video Camera. In *Proc. DocEng 2015*. 219–222.
4. A. Tang, C. Neustaedter, and S. Greenberg. 2007. Videoarms: embodiments for mixed presence groupware. *People and Computers* 20 (2007), 85–102.
5. Yuan Tian, Suraj Raghuraman, Thiru Annaswamy, Aleksander Borresen, Klara Nahrstedt, and Balakrishnan Prabhakaran. H-TIME: Haptic-enabled tele-immersive musculoskeletal examination. In *Proc. ACM Multimedia 2017*. 137–145.
6. E. Wood, J. Taylor, J. Fogarty, A. Fitzgibbon, and J. Shotton. ShadowHands: High-fidelity remote hand gesture visualization using a hand tracker. In *Proc. ISS 2016*. 77–84.
7. Ying Xiong. 2011. Fast and Accurate Document Detection for Scanning, Dropbox blogs. (2011). `https://blogs.dropbox.com/tech/2016/08/fast-and-accurate-document-detection-for-scanning/`
8. Christian Zimmermann and Thomas Brox. Learning to Estimate 3D Hand Pose from Single RGB Images. In *Proc. ICCV 2017*. 4913–4921.