Computer Vision Technologies for Remote Collaboration Using Physical Whiteboards, Projectors and Cameras

Zhengyou Zhang
Microsoft Research, One Microsoft Way
Redmond, WA 98052, USA
E-mail: zhang@microsoft.com

Abstract

A whiteboard provides a large shared space for the participants to focus their attention and express their ideas, and is therefore a great collaboration tool for information workers. However, it has several limitations; notably, the contents on the whiteboard are hard to archive or share with people who are not present in the discussions. This paper presents our work in developing tools to facilitate collaboration on physical whiteboards by using a camera and a microphone. In particular, we have developed two systems: a whiteboard-camera system and a projector-whiteboard-camera system. The whiteboard-camera system allows a user to take notes of a whiteboard meeting when he/she wants or in an automatic way, to transmit the whiteboard content to remote participants in real time and in an efficient way, and to archive the whole whiteboard session for efficient post-viewing. The projector-whitebord-camera system incorporates a whiteboard into a projector-camera system. The whiteboard serves as the writing surface (input) as well as the projecting surface (output). Many applications of such a system inevitably require extracting handwritings from video images that contain both handwritings and the projected content. By analogy with echo cancellation in audio conferencing, we call this problem **visual echo cancellation***, and we describe one approach to accomplish the task. Our systems can be retrofit to any existing whiteboard. With the help of a camera and a microphone and optionally a projector, we are effectively bridging the physical and digital worlds.*

1: Introduction

The work presented in this paper focuses on the particular meeting scenarios that use whiteboard heavily such as brainstorming sessions, lectures, project planning meetings, and patent disclosures. In those sessions, a whiteboard is indispensable. It provides a large shared space for the participants to focus their attention and express their ideas spontaneously. It is not only effective but also economical and easy to use – all you need is a flat board and several dry-ink pens.

While whiteboard sessions are frequent for knowledge workers, they are not perfect. The content on the board is hard to archive or share with others who are not present in the session. People are often busy copying the whiteboard content to their notepads when they should spend time sharing and absorbing ideas. Sometimes they put "Do Not Erase" sign on the whiteboard and hope to come back and deal with it later. In many cases, they forget or the content is accidentally erased by other people. Furthermore, meeting participants who are on conference call from remote locations are not able to see the whiteboard content as the local participants do. In order to enable this, the

meeting sites often must be linked with expensive video conferencing equipments. Such equipment includes a pan-tilt-zoom camera which can be controlled by the remote participants. It is still not always satisfactory because of viewing angle, lighting variation, and image resolution, without mentioning lack of functionality of effective archiving and indexing of whiteboard contents.

Our system was designed with three purposes:

1. to alleviate meeting participants the mundane tasks of note taking by capturing whiteboard content automatically or when the user asks;
2. to communicate the whiteboard content to the remote meeting participants in real time using a fraction of the bandwidth required if video conferencing equipment is used;
3. to archive the whole meeting in a way that a user (participants or not) can find efficiently the desired information.

In Sections 2 to 4, we describe the techniques used in each part. Because of space limitation, only a high-level description is provided.

To the best of our knowledge, all existing systems that capture whiteboard content in real time require instrumentation either in the pens or on the whiteboard. Our system allows the user to write freely on any existing whiteboard surface using any pen. To achieve this, our system uses an off-the-shelf high-resolution video camera which captures images of the whiteboard at 7.5Hz. From the input video sequence, our algorithm separates people in the foreground from the whiteboard background and extracts the pen strokes as they are deposited to the whiteboard. To save bandwidth, only newly written pen strokes are compressed and sent to the remote participants.

Furthermore, in order to facilitate interaction from remote users, we integrate a projector in the whiteboard-camera system. The projector can project annotations from remote users as well as contents from PowerPoint or Word documents. The whiteboard serves as the writing surface (input) as well as the projecting surface (output). In such a system, one vital requirement is the extraction of handwritings from video images that contain both handwritings and the projected content. By analogy with echo cancellation in audio conferencing, we call this problem *visual echo cancellation*. Section 5 describe the details of this system.

2: Whiteboard Scanning and Image Enhancement

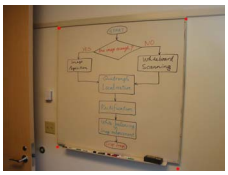
Because digital cameras are becoming accessible to average users, more and more people use digital cameras to take images of whiteboards instead of copying manually, thus significantly increasing the productivity. The system we describe in this paper aims at reproducing the whiteboard content as a faithful, yet enhanced and easily manipulable, electronic document through the use of a digital (still or video) camera.

However, images are usually taken from an angle to avoid highlights created by flash, resulting in undesired perspective distortion. They also contain other distracting regions such as walls. The system we have developed uses a series of image processing algorithms. It automatically locates the boundary of a whiteboard as long as there is a reasonable contrast near the borders, crops out the whiteboard region, rectifies it to a rectangle with the estimated aspect ratio, and finally correct the colors to produce a crisp image.

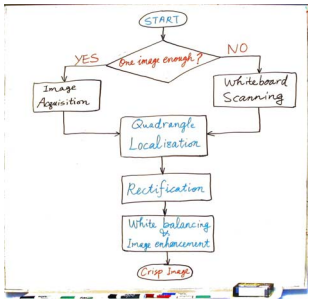
Besides image enhancement, our system is also able to scan a large whiteboard by stitching multiple images automatically. Our system provides a simple interface to take multiple images of the whiteboard with overlap and stitches them automatically to produce a high-res image. The stitched image can then be processed and enhanced as mentioned earlier.

2.1: Overview of the System

Let us take a look at Figure 1. On the top is an original image of a whiteboard taken by a digtal camera, and on the bottom is the final image produced automatically by our system. The content on the whiteboard gives a flow chart of our system.



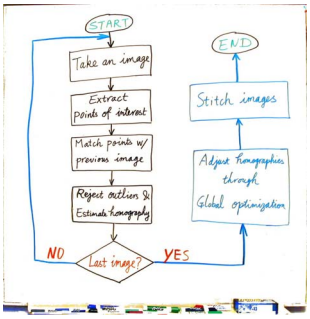
(a)



(b)

Figure 1. Diagram of the system architecture drawn on a whiteboard. (a) Original image; (b) Processed one.

(a)



(b)

Figure 2. Diagram of the scanning subsystem: (a) Original image; (b) Processed image.

As can be seen in Fig. 1b, the first thing we need to decide is whether it is enough to take a single image of the whiteboard. If the whiteboard is small (e.g., 40' by 40') and a high-resolution digital camera (e.g., 3 mega pixels) is used, then a single image is usually enough. Otherwise, we need to call the whiteboard scanning subsystem, to be described in Section 2.2, to produce a composite image that has enough resolution for comfortable reading of the whiteboard content. Below, we assume we have an image with enough resolution.

The first step is then to localize the borders of the whiteboard in the image. This is done by detecting four strong edges. The whiteboard in an image usually appears to be a general quadrangle, rather than a rectangle, because of camera's perspective projection. If a whiteboard does not have strong edges, an interface is provided for the user to specify the quadrangle manually.

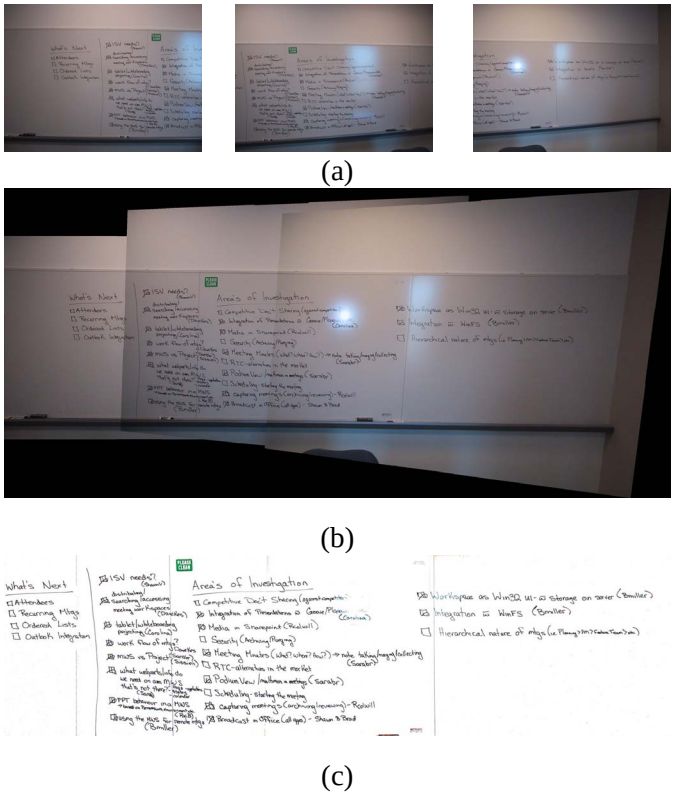
The second step is image rectification. For that, we first estimate the actual aspect ratio of the whiteboard from the quadrangle in the image based on the fact that it is a projection of a rectangle in space. From the estimated aspect ratio, and by choosing the "largest" whiteboard pixel as the standard pixel in the final image, we can compute the desired resolution of the final image. A planar mapping (a $3\times$ 3 homography matrix) is then computed from the original image quadrangle to the final image rectangle, and the whiteboard image is rectified accordingly.

The last step is white balancing of the background color. This involves two procedures. The first is the estimation of the background color (the whiteboard color under the same lighting without anything written on it). This is not a trivial task because of complex lighting environment, whiteboard reflection and strokes written on the board. The second concerns the actual white balancing. We make the background uniformly white and increase color saturation of the pen strokes. The output is a crisp image ready to be integrated with any office document or to be sent to the meeting participants.

Figures 1 and 2 each show the original image on the left and the processed image on the right.

2.2: Whiteboard Scanning Subsystem

The major steps of the Whiteboard Scanning system is illustrated in Figure 2. The mathematic foundation is that two images of a *planar* object, regardless the angle and position of the camera, are related by a plane perspectivity, represented by a 3×3 matrix called *homography* $\mathbf{H}$. The stitching process is to determine the homography matrix between successive images, and we have developed an automatic and robust technique based on points of interest. Due to space limitation, the reader is referred to [2] for details. An example of whiteboard scanning is shown in Fig. 3.



(a)

(b)

(c)

Figure 3. A second example of whiteboard scanning. (a) Three original images; (b) Stitched image; (c) Final processed image.

3: Real-time Whiteboard Processing and Collaboration

Sometimes, meeting participants who are on conference call from remote locations are not able to see the whiteboard content as the local participants do. In order to enable this, the meeting sites often must be linked with expensive video conferencing equipments. Such equipment includes a pan-tilt-zoom camera which can be controlled by the remote participants. It is still not always satisfactory because of viewing angle, lighting variation, and image resolution, without mentioning lack of functionality of effective archiving and indexing of whiteboard contents. Other equipment requires instrumentation either in the pens or on the whiteboard. Our system allows the user to write freely on any existing whiteboard surface using any pen. To achieve this, our system uses an off-the-shelf high-resolution video camera which captures images of the whiteboard at 7.5Hz. From the input video sequence, our algorithm separates people in the foreground from the whiteboard background and extracts the pen strokes as they are deposited to the whiteboard. To save bandwidth, only newly written pen strokes are compressed and sent to the remote participants.

3.1: Real-time Processing

The input to our real-time whiteboard system is a sequence of video images, taken with a high-resolution video camera. There are a number of advantages in using a high-resolution video camera over the sensing mechanism of devices like Mimio or electronic whiteboard. They are: 1) Without requiring special pens and erasers makes the interaction much more natural. 2) Since it is taking images of the whiteboard directly, there is no mis-registration of the pen strokes. 3) As long as

the users turn on the system before erasing, the content will be preserved. 4) Images captured with a camera provide much more contextual information such as who was writing and which topic was discussing (usually by hand pointing). However, our system has a set of unique technical challenges.

Since the person who is writing on the board is in the line of sight between the camera and the whiteboard, he/she often occludes some part of the whiteboard. We need to segment the images into foreground objects and whiteboard. For that, we rely on two primary heuristics: 1) Since the camera and the whiteboard are stationary, the whiteboard background cells are stationary throughout the sequence until the camera is moved; 2) Although sometimes foreground objects (e.g., a person standing in front of the whiteboard) occlude the whiteboard, the pixels that belong to the whiteboard background are typically the majority.

We apply several strategies in our analysis to make the algorithm efficient enough to run in real time.

First, rather than analyzing the images at pixel level, we divide each video frame into rectangular cells to lower the computational cost. The cell size is roughly the same as what we expect the size of a single character on the board (16 by 16 pixels in our implementation). The cell grid divides each frame in the input sequence into individual cell images, which are the basic unit in our analysis.

Second, our analyzer is structured as a pipeline of six analysis procedures (see Figure 4). If a cell image does not meet the condition in a particular procedure, it will not be further processed by the subsequent procedures in the pipeline. Therefore, many cell images do not go through all six procedures. At the end, only a small number of cell images containing the newly appeared pen strokes come out of the analyzer.

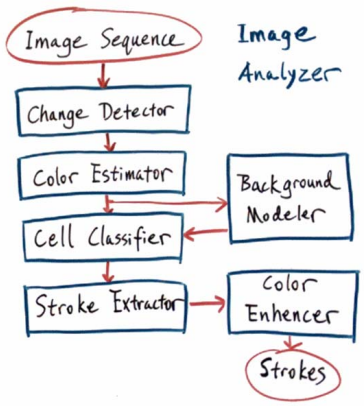


Figure 4. The image sequence analysis process

The third strategy is specific to the video camera, Aplux MU2, that we use in our system. The Aplux MU2 allows the video frames to be directly accessed in Bayer format, which is the single channel raw image captured by the CMOS sensor. In general, a demosaicing algorithm is run on the raw image to produce an RGB color image. By processing the cell images in raw Bayer space instead of RGB space and delaying demosaicing until the final step and running it only on the cells containing new strokes, we save memory and processing by at least 66%. An additional benefit is that we can obtain a higher quality RGB image at the end by using a more sophisticated demosaicing algorithm than the one built into the camera driver.

3.2: Teleconferencing Experience

We have implemented our system as a plug-in to the Whiteboard applet of the Microsoft Windows Messenger (see Figure 5). The Whiteboard applet allows the users at two ends of a Windows Messenger session to share a digital whiteboard. The user at one end can paste images or draw geometric shapes and the user at the other end can see the same change almost instantaneously. Usually, the user draws objects with his mouse, which is very cumbersome. With our system, the user can write on a real whiteboard instead.

The changes to the whiteboard content are automatically detected by our system and incrementally piped to the Whiteboard applet as small cell image blocks. The Whiteboard applet is re-

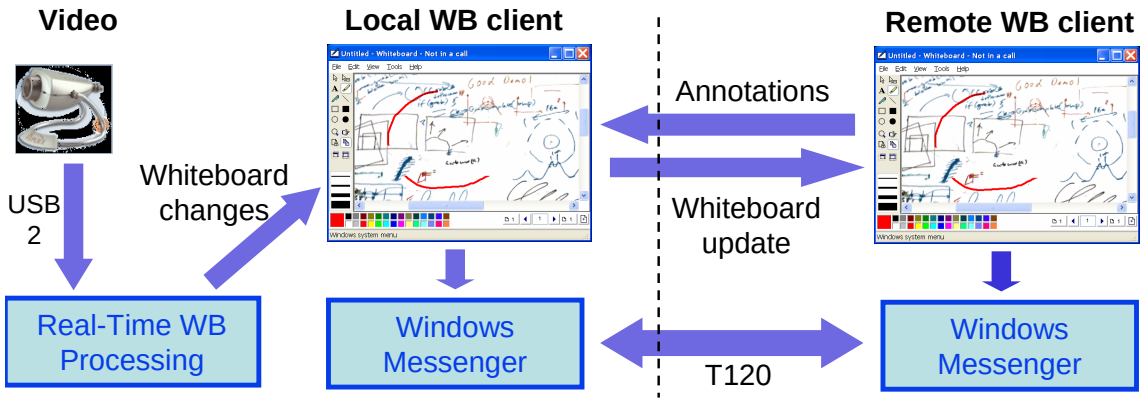


Figure 5. Real-time whiteboard system inside the Windows Messenger

sponsible for compressing and synchronizing the digital whiteboard content shared with the remote meeting participant. The remote participant can add annotations on top of the whiteboard image using the mouse. When used with other Windows Messenger tools, such as voice conferencing and application sharing, whiteboard sharing becomes a very useful tool in communicating ideas.

A video capturing the working of our real-time whiteboard system will be shown during the conference. The reader is referred to [3] for details.

4: Whiteboard Archiving

As with the previous system, we point a high-resolution digital still camera to the whiteboard and continuously capture its images. Additionally, we also capture the audio discussions with a microphone. During the post-processing stage, our system distills a small set of key frame images from the captured image sequence. A key frame represents the maximum content on the whiteboard before each erasure. Time stamps of the pen strokes contained in the key frames are also computed. The users can view the key frame images, print them as notes, or cut and paste them into documents. If the users want to find more about the discussion on a particular topic, our browsing software allows them to click some pen stroke associated with that topic and bring up the audio at the moment when the stroke was written. Therefore the whiteboard content serves as a visual index to efficiently browse the audio meeting.

4.1: Browsing Interface

Since most people probably do not want to listen to the recorded meeting from start to end, we provide two browsing features to make non-linear accessing of the recorded information very efficient (see Fig.6 and its caption). 1. Key Frames: Key frame images contain all the important content on the whiteboard and serve as a summary to the recording. They can be cut and pasted to other documents or printed as notes. 2. Visual Indexing: We provide two levels of non-linear access to the recorded audio. The first is to use the key frame thumbnails. The user can click a thumbnail to jump to the starting point of the corresponding key frame. The second is to use the pen strokes in each key frame. Together with the standard time line, these two levels of visual indexing allow the user to browse a meeting in a very efficient way.

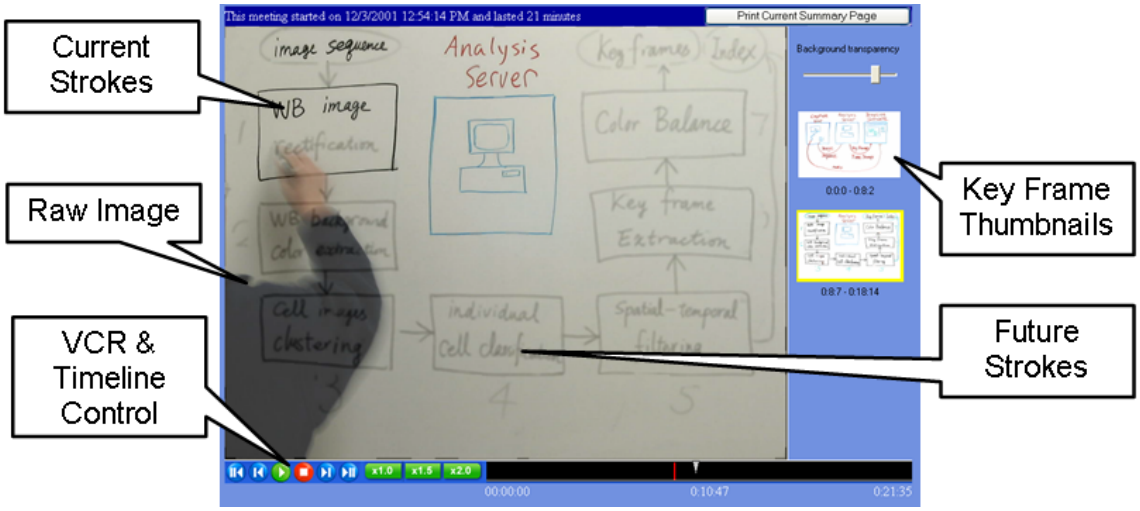


Figure 6. Browsing interface. Each key frame represents the whiteboard content of a key moment in the recording. The main window shows a composition of the raw image from the camera and the current key frame. The pen-strokes that the participants are going to write in the future (Future Strokes) are shown in ghost-like style.

4.2: Technical Challenges

The input to the Whiteboard Capture System is a set of still digital images (see Figure 2). We need to analyze the image sequence to find out when and where the users wrote on the board and distill a set of key frame images that summarize the whiteboard content throughout a session. Compared to other systems, our system has a set of unique technical challenges: 1) The whiteboard background color cannot be pre-calibrated (e.g. take a picture of a blank whiteboard) because each room has several light settings that may vary from session to session; 2) Frequently, people move between the digital camera and the whiteboard, and these foreground objects obscure some portion of the whiteboard and cast shadow on it. Within a sequence, there may be no frame that is totally un-obscured. We need to deal with these problems in order to compute time stamps and extract key frames.

Rather than analyzing images on per-pixel basis, we divide the whiteboard region into rectangular cells to lower computational cost. The cell size is roughly the same as what we expect the size of a single character on the board (about 1.5 by 1.5 inches). Since the cell grid divides each frame in the input sequence into cell images, we can think of input as a 3D matrix of cell images.

Figure 7 shows an outline of the algorithm. Due to space limitation, the reader is referred to [1] for details. A demo will be shown during the conference.

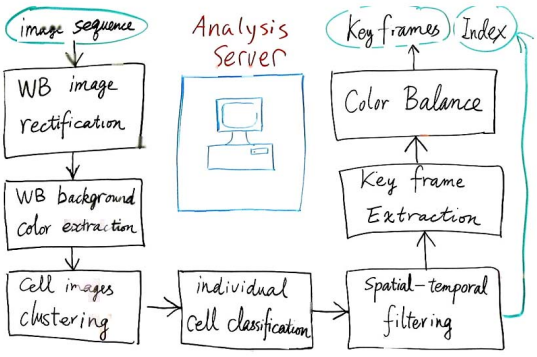


Figure 7. The image sequence analysis process

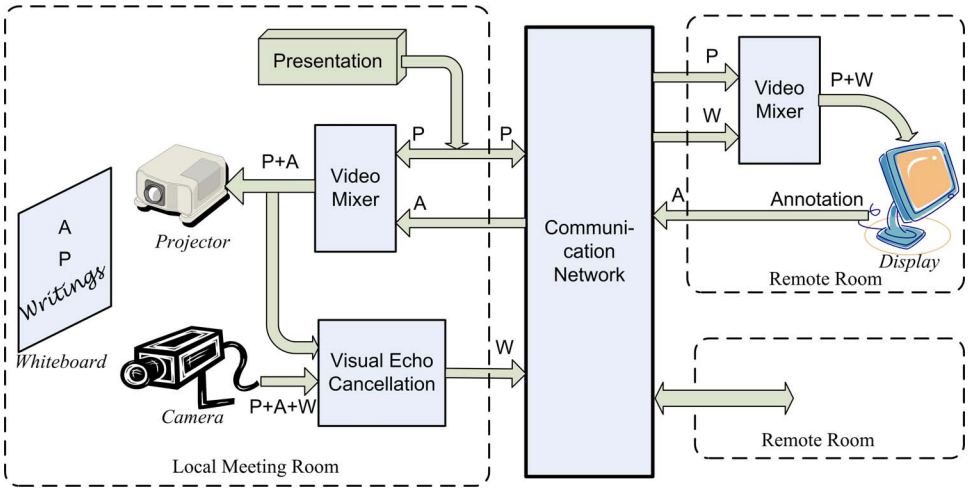


Figure 8. A projector-whiteboard-camera system

5: A Projector-Whiteboard-Camera System

During the past few years we witnessed the transformation of video cameras and projectors from expensive lab equipments to affordable consumer products. This triggers many human-computer interaction systems that incorporate both the large-scale display provided by the projector and intelligent feedback from one or more cameras. On the other hand, the whiteboard is still an indispensable part of many meetings (including lecturing, presentation and brainstorming), because it provides a large shared space for the participants to focus their attention and exchange their ideas spontaneously. One can write or draw his/her idea on it with an easily accessible marker. Therefore we propose to integrate the whiteboard into a projector-camera system by using it as both the writing surface and the projecting surface. Several immediate advantages are:

1. Computer presentations (such as PowerPoint) and whiteboard discussions are seamlessly integrated into one session. Meeting attendees will not be distracted by switching from the screen to the whiteboard, and vice versa.
2. Such a system enables local and remote attendees to collaborate with each other on a shared workspace. Local attendees have a much more natural writing surface than most commercial large display products.
3. Most importantly, the system can be easily deployed on top of current meeting environments. It is therefore much more economical than most large display products that requires installing expensive equipments and accessories.

Since the captured video contains both writings on the physical whiteboard and contents projected from the computer, it is very important to separate whiteboard writings from the projected contents. Some of the benefits from this separation are:

1. It dramatically reduces the bandwidth requirement for teleconferencing, because both extracted writing and the computer-projected contents can be transmitted with very low bandwidth, comparing with the original mixed video, since the video is affected by shadow and lighting variation.
2. It considerably improves the remote users experience in teleconferencing in several ways, to be discussed below.

3. Extracted writings are essential for archiving and browsing meetings offline. Writing on the whiteboard usually indicates an important event in a meeting.
4. By feeding the results to an OCR (Optical Character Recognition) system, the meeting archive can be more easily accessed and transferred into other forms.

By analogy with echo cancellation in audio conferencing, we call this problem *visual echo cancellation*. *Visual Echo*, by strict definition, is the appearance of the projected contents viewed by the camera. *Visual Echo Cancellation* is defined as extracting the physical writings from the video containing both the writings and the visual echoes. In order to achieve this goal, we need an accurate prediction of the appearance of the computer projected content as viewed by the camera. This requires two basic components:

1. Geometric calibration: It concerns the mapping between the position in the camera view and the position in the projector screen.
2. Color calibration: It concerns the mapping between the actual color of the projected content and that seen by the camera.

For geometric calibration, we assume that both camera and projector are linear projective, and implement a robust, accurate and simple technique by leveraging the fact that the projector can actively project the patterns we want. For color calibration, we model pixels on the visual echo as independent Gaussian random variables and propose a lookup-table-based approach. Note that both components are useful for other projector-camera systems.

5.1: System Overview

Figure 8 illustrates how our projector-camera-whiteboard system works. The local meeting room is equipped with a projector, a camera, and a whiteboard. The projector and the camera are rigidly attached to each other, although theoretically they can be positioned anywhere as long as the projector projects on the whiteboard and the camera sees the whole projection area. The projector and the camera are linked to a computer, and the computer is connected to the communication network. Remote attendees also connect their computers to the communication network.

A presentation could be PowerPoint slides, a spreadsheet, a PDF file, etc. The data stream for the presentation is indicated by "`P`" in the figure. Remote attendees may annotate the presentation, and the annotation stream is indicated by "`A`". Both "`P`" and "`A`" are mixed together before sending to the projector for projecting on the whiteboard. During the presentation, the presenter or other local attendees may write or draw on the whiteboard. The camera captures both the projected content and the writings. Through geometric and color calibrations, the system predicts the appearance of the projected "`P`" and "`A`" viewed by the camera, i.e., the visual echo. The *Visual Echo Cancellation* module tries to extract only the writings on the whiteboard, indicated by "`W`", by subtracting the predicted visual echo from the live video. At the remote side, the presentation stream "`P`" and the whiteboard writing stream "`W`" are mixed before displaying on the computer.

5.2: Geometric Calibration

For visual echo cancellation, we need to know the relationship between the position in the camera view and the position in the projector screen. This is the task of geometric calibration. Assuming that both camera and projector are linear projective and that the whiteboard surface is planar, it can be easily shown that the mapping between a point in the camera view and a point in the projector screen is a homography, and can be described by a 3×3 matrix $\mathbf{H}$ defined up to a scale factor.

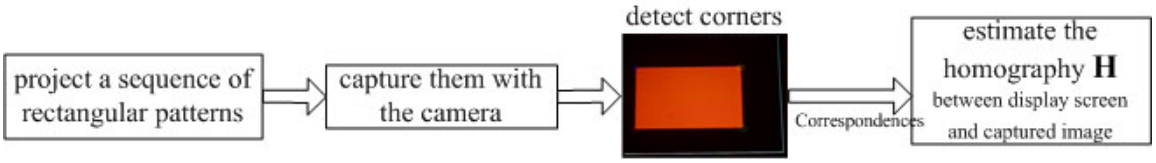


Figure 9. Flowchart for geometric calibration

Figure 9 shows the flowchart for geometric calibration. The idea is to leverage the fact that the projector can actively project the patterns we want. The whole process takes less than 2 minutes and is only necessary when camera is moved with respect to the projector. The main steps are:

1. Sequentially project N ($N = 40$ in our implementation) rectangles and simultaneously capture their images using a fixed camera.
2. Detect the 4 corners of each rectangle in the images.
3. Use the $4 \times N$ detected corners and their corresponding known positions in the projector space to estimate the homography between the projector screen and the image plane of the camera.

Note that in theory, only 4 points (i.e., one rectangle) are necessary to estimate the homography. In order to achieve higher accuracy, we use more rectangles and they are projected in different locations of the whiteboard.

Our method takes advantage of the fact that the relative position between the camera and the projection surface is fixed during the calibration. Therefore correspondences detected in different images can be used for estimating a single homography, which increase the accuracy and robustness of our method without complicating the corner detection algorithm.

The corner detection process consists of the following main steps:

1. Convert color images to grayscale images.
2. Detect edges on the grayscale image.
3. Use hough transform to detect straight lines on the edge map.
4. Fit a quadrangle using the lines.
5. Find the corners of the quadrangle.

In order to reduce the noise in edge map, we need to find the region inside and outside the rectangle and quantize the grayscale value. Since the inside region is bright and homogeneous, it forms peak $p1$ at the higher range of the histogram, while the background forms peak $p2$ at the lower range. We use a coarse-to-fine histogram-based method to find the two peaks and set the higher threshold $h1 = \frac{3}{4} \times p1 + \frac{1}{4} \times p2$ and the lower threshold $h2 = \frac{1}{4} \times p1 + \frac{3}{4} \times p2$. The grayscale level of all pixels above $h1$ are set to $h1$, while those below $h2$ are set to $h2$, and those in between remain unchanged.

5.3: Color Calibration

For visual echo cancellation, for a given pixel in the projector space, we know its corresponding position in the camera space through geometric calibration described above; furthermore, we need to know what the corresponding color should look like in the captured video, and this is the task of color calibration. Note that the same color in the projector space appears different in the camera, depending where the color is projected on the whiteboard. This is because the projector lamp does not produce uniform lights, the lighting in the room is flickering and not uniform, and

the whiteboard surface is not Lambertian. Therefore, color calibration should be both color- and position-dependent.

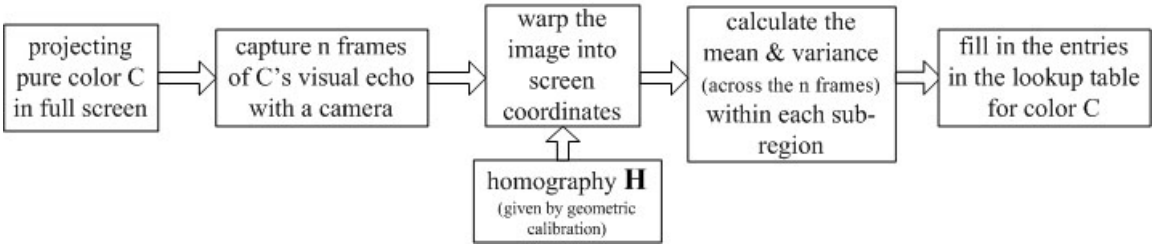


Figure 10. Flowchart for color calibration

Figure 10 shows the flowchart for color calibration. Below are the main steps:

1. Quantize the RGB color space into $9 \times 9 \times 9 = 729$ bins.
2. Project each quantized color over the whole display region and capture its image in synchronization. We store n ($n = 5$) frames for each color.
3. Rectify using the geometric calibration and divide the display region evenly into $32 \times 32 = 1024$ rectangular blocks.
4. Calculate the mean and variance of each color in each block across the n frames.

In this way, we build a lookup table for the 729 quantized colors at each of the 1024 blocks. Note that the spatial dimensionality is necessary because the same projected color will have different appearance at different position, as the second row in Figure 12 shows. The best result would be obtained if a lookup table were built for every pixel position, but it seems unnecessary from our experiments because the color appearance changes smoothly across the display surface.

Given arbitrary display content, we estimate the *visual echo* E by:

1. Substitute each pixel with its correspondent mean color in the lookup table[1].
2. Backward-warp it to the camera view. To obtain an estimate of the error bound for each pixel, we also lookup and warp the variances to get a pixel-wise variance map V.

5.4: Visual Echo Cancellation

Let us now look at visual echo cancellation. Figure 11 shows the flowchart of the cancellation process. The details are explained in the following subsections.

5.4.1: Generative Process of The Captured Image

By writing/drawing with a paint marker on the whiteboard, we actually change the surface albedo of the whiteboard, and therefore change the reflection. Therefore extracting the writings boils down to detecting the changes on the surface albedo.

Assuming all the images are geometrically aligned, and denoting the incident light map by P, the surface albedo of the whiteboard by A, the pixel-wise color transformation due the camera sensor by C, and the visual echo by E, we have $E = C \times A \times P$. If nothing is written on the whiteboard, then the captured image I should be equal to E. If there is any thing written on the whiteboard, the surface albedo changes, and is denoted by $\tilde{A}$. The captured image can then be described by

[1]For colors not in the table, we use linear interpolation of the two nearest bins.

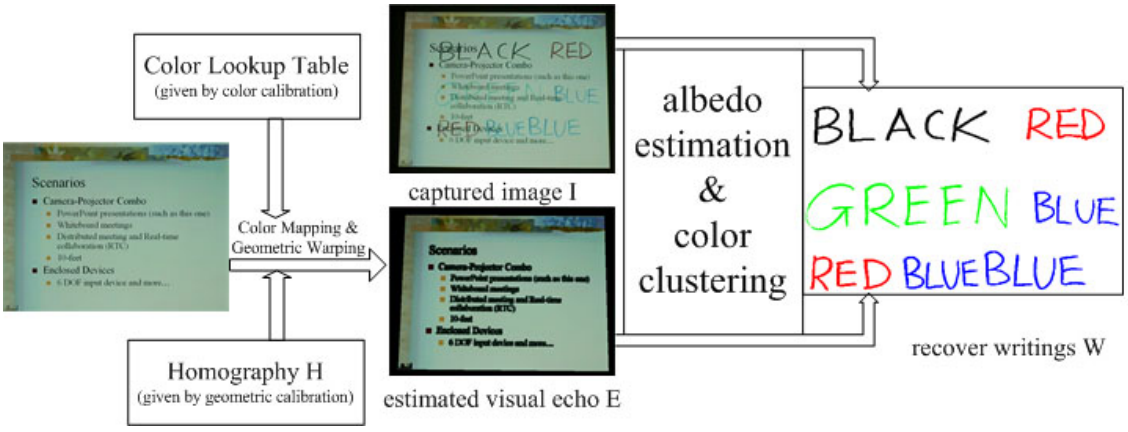


Figure 11. Flowchart for visual echo cancellation

$I = C \times \tilde{A} \times P$. We can compute the albedo change by estimating the *albedo ratio* $a = \tilde{A}/A$ of the pixel $[x, y]$ in color channel $c \in \{R, G, B\}$, which is given by

$$a_{[x,y],c} = \frac{I_{[x,y],c}}{E_{[x,y],c}} \tag{1}$$

Note that writings on the whiteboard absorb the lights, so $\tilde{A} \leq A$, and in consequence $a_{[x,y],c} \leq 1$.

Based on the albedo ratio a, we can detect the writings and recover its color. The albedo for the whiteboard region without writings should be 1. Assuming the sensor noise on the albedo is additive and has a zero-mean Gaussian distribution with variance $\frac{V}{E}$, we have the following decision rule:

Pixel $[x, y]$ belongs to the written region if and only if

$$1 - \frac{a_{[x,y],R} + a_{[x,y],G} + a_{[x,y],B}}{3} > \frac{V_{[x,y],R} + V_{[x,y],G} + V_{[x,y],B}}{E_{[x,y],R} + E_{[x,y],G} + E_{[x,y],B}} \tag{2}$$

Note that the decision rule is one-sided, because, as mentioned earlier, the albedo ratio for written whiteboard region is strictly less than 1.

For each pixel $[x, y]$ belongs to the written region, we can recover the writings with its colors as

$$W_{[x,y],c} = a_{[x,y],c} \times 255 \tag{3}$$

assuming the color intensity ranges from 0 to 255.

5.4.2: Practical Considerations

Due to the noise in geometric calibration, I and E are not exactly aligned. The 1 to 2 pixel errors are most evident near strong edges in E. Therefore in written region segmentation, we first apply an erosion on E, which increases the dark region. Thus the pixels near the dark regions in E have higher A and are less likely be classified as written region. This preprocessing reduces error because in order to make their writings more visible, most users prefer to write on top of brighter background instead of darker background.

In practice, to make the colors in W visible, we need to set the camera exposure to be much higher than normality. This will cause over-exposure during color calibration. We address this problem by setting the exposure optimal for color calibration, and use a classification method to

recover the colors of the writings. We choose the four most commonly used markers (red, black, blue and green) as classes $M_0 \sim M_3$. For supervised training, we use Equations (2) and (3) to recover a set of writings W, and then convert it from RGB color space to HSI (hue, saturation and intensity) color space, and denoting the new image as W'. We label the training data for class M_i by manually by selecting the region of written by Marker i, and collect its histogram $n_i(h, s, i)$.

To classify a pixel $W_{[x,y]}$ obtained from Equation (3), we convert its RGB value to $W'_{[x,y]}$ in HSI space and evaluate the likelihood that it belongs Cluster i ($i = 0, \ldots, 3$) as

$$p([x, y]|M_i) = \frac{n_i(W'_{[x,y],h}, W'_{[x,y],s}, W'_{[x,y],i})}{N_i}, \tag{4}$$

where N_i is the total number of data points in Histogram i.

Due to noise in camera sensor, a MAP decision rule may not give spatially consistent results, so we use a 61×61 window the collect votes from all the pixels in the neighborhood and classify the center pixel based on the maximum votes.

5.5: Experimental Results

We tested our geometric calibration method using various projectors (including an InFocus LP530 and a Proxima DP6155) and various video cameras (including a Aplex USB2, a Logitech Pro4000 and a SONY EVI30), under both artificial lighting and natural lighting conditions. The fitting error for solving the homography based on correspondences ranges from 0.3 to 0.7 pixels.

For color calibration, we use a SONY projector and EVI30 camera. Comparing the estimated visual echo E with the actual captured image I, the average error is around 3 (color intensity range $0 \sim 255$). The majority of the discrepancy is around the regions with strong edges, due to the noise in geometric calibration.

Figure 12. Experimental results for Visual Echo Cancellation

Figure 12 shows the visual echo cancellation results on various backgrounds. One can see that majority of the writings are recovered except for the parts on top of the extreme complex background contents. In the latter, even human eyes can hardly tell anyway.

6: Conclusions

We have described a whiteboard-camera system and a projector-whiteboard-camera system.

In the whiteboard-camera system, we have developed various computer vision technologies to increase the productivity in using physical whiteboards. In particular, "Whiteboard Scanning" captures notes on a whiteboard by taking one or multiple snapshots, thus relieving the participants of a meeting from the burden of copying the contents manually; "Real-Time Whiteboard" allows the users to share ideas on a whiteboard in a variety of teleconference scenarios such as brainstorming sessions, lectures, project planning meetings and patent disclosures, but only takes a fraction of its bandwidth and is suitable even on dial-up networks; "Whiteboard Archiving" records both whiteboard activities and audio signals, and helps the participants to review the meeting efficiently at a later time by providing key frame images that summarize the whiteboard content and structured visual indexing to the audio. The system has been tested extensively, and excellent results have been obtained.

We have also defined the problem of visual echo cancellation in a projector-whiteboard-camera system and proposed a solution using both geometric calibration and color calibration. Visual echo cancellation has wide applications in real-time collaboration tasks, both on-site and remotely. The algorithm is tested on various backgrounds and display contents, and good results are achieved.

Both geometric and color calibrations could be used for other purposes. The geometric calibration technique has actually been integrated into our camera-projector based human-computer-interaction system, which tracks the image position of the laser dot to command the mouse cursor on the display screen.

Some of the limitations of our current camera-projector-whiteboard system are:

1. Most whiteboard surfaces are not designed for projecting screen. Thus they give more specular highlights than regular projector screens. To avoid the consequent glaring effect to the audience, the projector should be posed at a large (either from very high or from very low) angle with respect to the whiteboard.
2. We need to redo color calibration if certain settings for the projector (color temperature, contrast or brightness) or the camera (exposure or white balance) are changed. Projecting and capturing $729 \times n$ ($= 3645$ when $n = 5$) frames at 10 fps (to ensure projecting and capturing are synchronized) takes about 6 minutes.

Acknowledgement. The authors are grateful to Li-wei He, Zicheng Liu and Hanning Zhou for their contributions and discussions in developing the systems described in this paper. Previous publications are listed below.

References

[1] L. He, Z. Liu, and Z. Zhang, "Why take notes? use the whiteboard system," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, Hong Kong, Apr. 2003, vol. V, pp. 776–779.

[2] Z. Zhang and L. He, "Notetaking with a Camera: Whiteboard Scanning and Image Enhancement", in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*, May 17-21, 2004, Montreal, Quebec, Canada.

[3] L. He and Z. Zhang, "Real-Time Whiteboard Capture and Processing Using a Video Camera", in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP 2005), Vol.2, pp.1113-1116, March 18-23, 2005, Philadelphia.

[4] H. Zhou, Z. Zhang, and T.S. Huang, "Visual Echo Cancellation in a Projector-Camera-Whiteboard System", in *Proc. International Conference on Image Processing* (ICIP), Vol. 5, pp. 2885–2888, Oct.24–27, 2004, Singapore.