A Computer Vision based Whiteboard Capture System

Richard Yi Da Xu
Center for Research in Complex Systems
Charles Sturt University
rxu@csu.edu.au

Abstract

Conventional whiteboard video capture using a static camera usually results in a poor quality. In this paper, we present an autonomous whiteboard scan and capture prototype system, which consist a pair of static and Pan-Tilt-Zoom (PTZ) cameras. The PTZ camera is used to scan the newly-updated whiteboard regions without interrupting the instructor. We will illustrate several computer vision techniques used in our system: Firstly, we present our unique camera calibration method using rough hand-drawn gridlines. Secondly, we present the image processing methods used to determine where the newly updated whiteboard region to be scanned is. Our method also accounts for the whiteboard region occlusion from the instructor.

1. Introduction

In order to digitally record and transmit the lecture information, many teaching institutions are currently using *electronic whiteboard* to capture the instructor writings. However, electronic whiteboards are expensive and bulky. They also have fixed resolution and cannot capture real objects, for example, a poster stick on the whiteboard. Video recording the whiteboard changes, on the other hand, is a cost-effective alternative solution. However, if a static camera is placed to oversee the entire whiteboard, then the writings captured from it often have poor readability. In addition, when the whiteboard is small in size, the instructor often obstructs part of the writing regions, making it difficult for the remote students to read.

Recently, many image processing and computer vision methods are used to automate and enhance the traditional whiteboard capture process. The early attempts are mostly based on occasional snapshot-taking [1]. Recognizing the importance of continuous whiteboard information capturing, the authors in [2] proposed a system, where the writing process is continuously observed from a static video camera positioned close to a moderate-size whiteboard. In the recent Microsoft RTWCS project [3, 4], the authors have proposed a system which uses the off-the-shelf camera for whiteboard capturing. The system is able to reconstruct and enhance the whiteboard images to a uniform background.

We designed an autonomous, real-time whiteboard capture prototype system which aims to continuously display the rectified and enhanced whiteboard images to the remote participants. The enhanced whiteboard view is at a higher resolution than those offered in the previous works. In order to capture the high resolution whiteboard images, our system uses a PTZ camera to zoom up-close and scan the newly written whiteboard region. At the same time, we use a static camera to continuously monitor the events occurred and makes decision on location of the next scanning region. This process is achieved without human intervention, and will not interfere with an instructor's normal presentation routine.

Our prototype system uses several computer vision and camera control methods. In order to present the system and its underlying techniques, the rest of this paper is organized as follows: In section 2, we describe how we calibrate the whiteboard views between the static and PTZ camera using rough hand-drawn gridlines. In section 3, we will describe how the system robustly distinguishes between whiteboard regions with and without writings and whether the region is occluded by the instructor. In section 4, we will describe the whiteboard image transform and enhancement.

2. Cameras Calibration Using Hand-Drawn Gridlines

The hardware used in our system is comprised of a Sony EVI-D100 PTZ camera and a Logitech 4000 static camera. Both cameras are placed close to each other, and at the same time, the static camera can view the entire whiteboard. All the image processing and camera control software is running on a standard PC. The hardware configurations are shown in fig 1:

2.1. Hand-drawn quadrilateral mesh detection

Figure 1: the hardware configuration

When the PTZ camera is zoom close-up, we must know where the corresponding region in the static camera view is, and vice versa. Therefore, a calibration between the static camera view and each of the zoom close-up view is required.

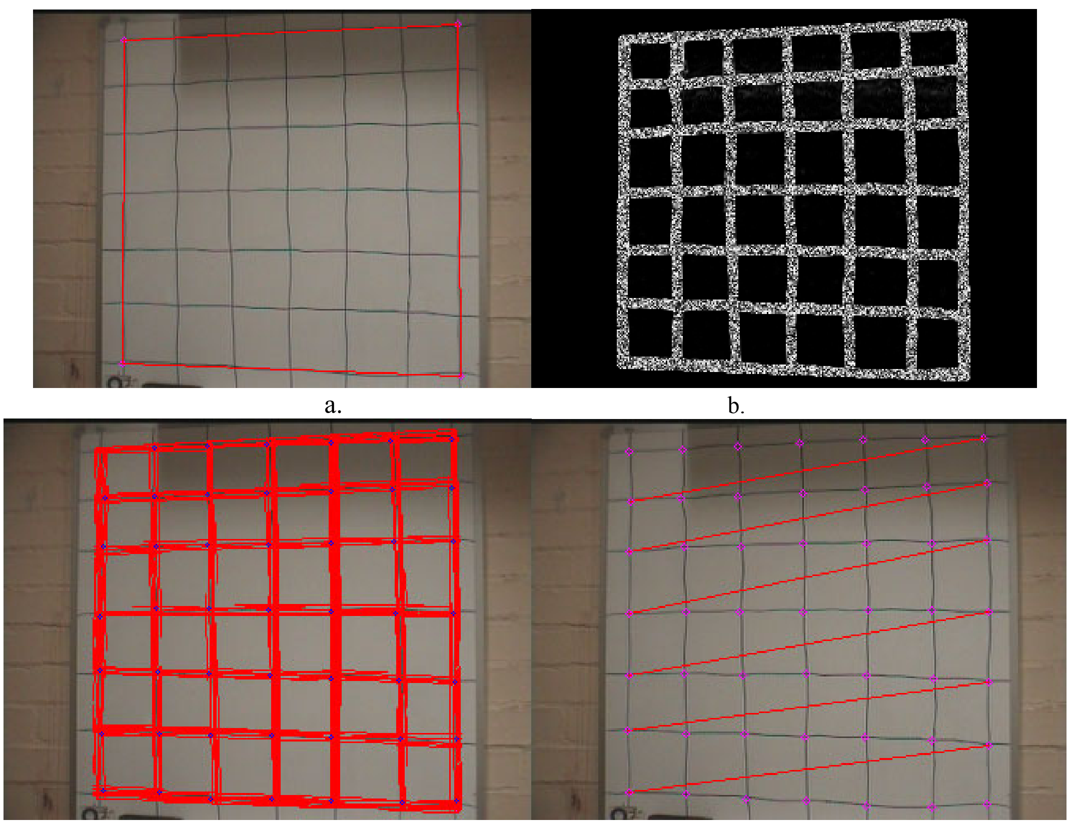
We have designed two effective and automated whiteboard quadrilateral mesh detection methods. In our earlier approach, all user needs to do is to hand draw a set of gridlines on the whiteboard (fig 2.a), which they can be very rough straight lines. We then applied several image processing methods, in order to robustly detect the corners of each grid, which forms a quadrilateral mesh.

We use Harris edge detector to detect the edges (fig 2.b) within the specified whiteboard region. We then apply probabilistic Hough Transform to the edge image, to obtain a set of lines (fig 2.c). By using the intersection angles criteria (we use |cos(θ| < cos(70)), we can predicate all the corner positions.

Because the gridlines are only roughly straight, in addition to the camera noises and shadows, therefore, the estimated corners differ from the true ones. However, a set of search windows can be formed where the estimated corners are their centers. Within each search window, we compute the true corner by using equation 1:

$$q = (\sum \nabla I_{p_i} . \nabla I_{p_i}^{\ T}).(\sum \nabla I_{p_i} . \nabla I_{p_i}^{\ T} . p_i) \qquad (1)$$

where ∇I_{p_i} is the image gradient at one of the points P_i in a neighborhood of q. The result is shown in fig 2.d:



a. b.

c. d.

Figure 2: Quadrilateral mesh detection for calibration - earlier approach (a) Manually select the outmost whiteboard region (b) Harris edge detection (c) line detection using probabilistic Hough Transform. (d) Detected corners.

Our system can also estimate the whiteboard aspect ratio and detect whiteboard regions automatically. However, in practice, we find it is much easier and more robust for user to manually specify these values. In addition, we usually ask the user to specify the whiteboard region using the four outermost grid corners, rather than the four true whiteboard corners. In this way, we have a much robust corners tracking (section 2.2) result, with insignificant reduction in writing area.

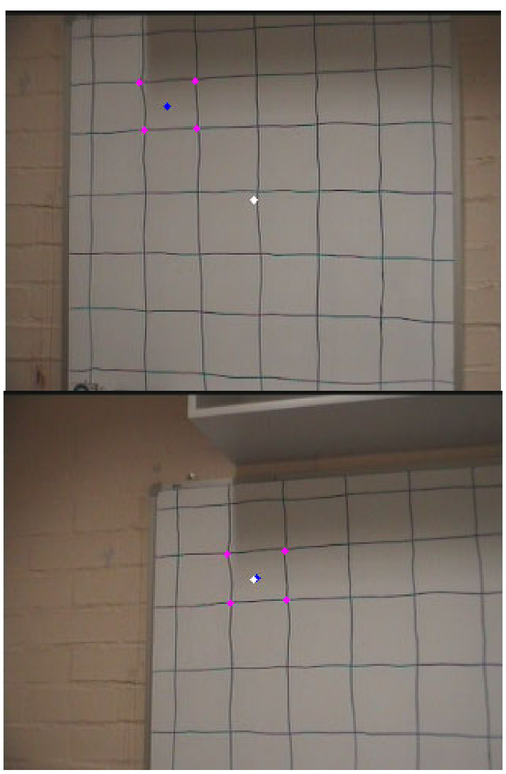
For the static camera view, once all the mesh corner points are detected, it is saved into an XML file.

2.2. Obtain every PTZ camera coordinates at zoom close-up

For the PTZ camera, when it is zoom close-up at each quadrilateral region, its pan, tilt and zoom value will change.

In our work, we determine and record these values automatically using corner tracking.

During this process, the PTZ camera is programmed to continuously changes its pan and tilt values until the distance between the center of the quadrilateral (blue dot in fig 3) and camera image center (white dot in fig 3) is less than λpt. The zoom operation stops when the enclosed quadrilateral size is greater than $\lambda zoom$ times the image frame size. We normally set the value of $\lambda zoom$ to be between 0.1 and 0.2. The process repeats for every quadrilateral.



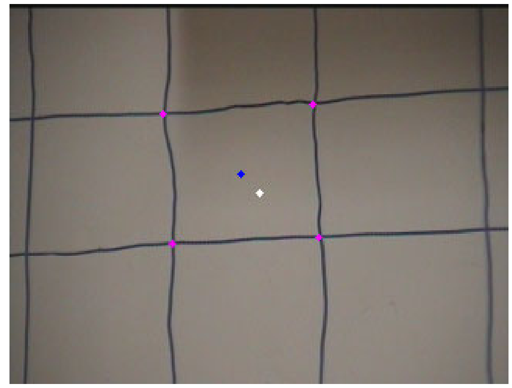


Figure 3: the image sequence of the PTZ camera view, when it is zoom close-up to a quadrilateral, with its corners being continuously tracked.

In order to track the quadrilateral corners when PTZ camera undergoes pan, tilt and zoom operations, we use the same corner search method described in section 2.1. Only this time, the center of the search window is taken from the previous detected corner position and the size of the search window is a circle with radius proportional to the smallest quadrilateral side length. An image sequence, showing quadrilateral tracking is shown in fig 3. A full length video showing the entire calibration process (7x7 hand-drawn grids) can be viewed at:

http://silica.csu.edu.au/staff/cs/rxu/videos/ptzscan_1.wmv

When the calibration process finishes, the complete pan-tilt-zoom values for each quadrilateral at zoom close-up are stored in an XML file.

At this time, the grid lines can be erased from the whiteboard. This calibration process only needs to repeat if the camera or the whiteboard position is changed.

2.3. An alternative method to whiteboard corner detection:

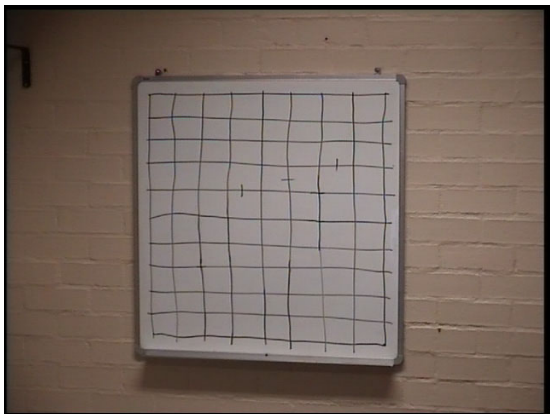
Recently, we have designed an improved technique to the whiteboard quadrilateral mesh detection. In this new method, the user is to specify three corners: the leftmost corner and its immediate horizontal and vertical neighboring corners.

Then, from the Laplace of Gaussian (LoG) filtered whiteboard image, the system detects the next corner recursively using a Gaussian-weighted predication line tracking technique, such that the next horizontal pixel location x_n is calculated using:

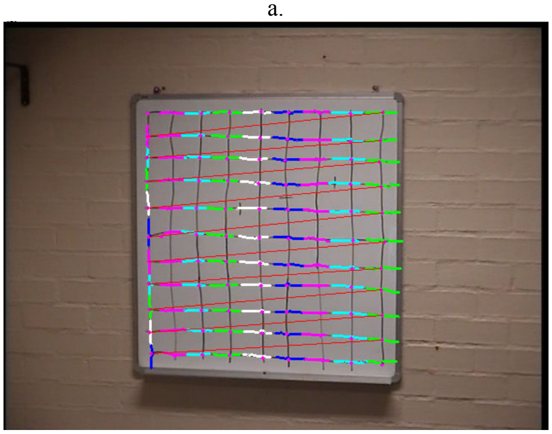
$$x_n = \underset{x \in [x_{n-1} + \frac{1}{2}(x_{n-1} - x_{n-2}) : x_{n-1} + \frac{3}{2}(x_{n-1} - x_{n-2})]}{\arg\max} \int_{x \in W} G(x) \bullet R(x) \qquad (2)$$

where $G(x) = Ne^{1/2\left[(x-(2x_{n-1}-x_{n-2}))/\sigma\right]^2}$

In equation 3, x_{n-1} and x_{n-2} are the two previous detected corners and σ is the chosen variance and N is the normalization constant. W is a window size, typically 3 or 5. When it's used to detect the next vertical grid corner, the symbol x_n is replaced by y_n. $R(x)$ the measurement of pixel variation in y-direction (in the case of vertical grid corner detection, it's the x-direction). Our method is robust against significant camera lens radial distortion, noises, projective distortion, and un-straight lines, such as the whiteboard image shown in fig 4.



a.



b.

Figure 4: process of grid corner detection using the newer method (a) the whiteboard image (b) detected grid corners and showing each searched length segment

For the PTZ camera, once the initial four corner points are detected, the PTZ camera will automatically undergo pan and tilt operation, until the four corners are located at the center of its image view before it moves to the next four corners. In this method, we use a constant zoom-level, and the demo of this work can be seen on: http://silica.csu.edu.au/staff/cs/rxu/videos/ptzscan_2.wmv

3. Whiteboard Scan Event Detection

By processing the images from the static camera view continuously, the system can determine when and which quadrilateral the PTZ camera needs to zoom close-up. In order for a scan to occur, two criteria must satisfy simultaneously: firstly, a whiteboard quadrilateral has updated and secondly, it is not occluded by the instructor.

Robust detections of these events are challenging, due to the room lighting and camera noises. In addition, the moving person can cast shadows. Therefore, a statistical whiteboard background modeling and updating method must be incorporated. In our work, we have used a two-step approach, aiming to improve the robustness. The first step is to obtain a set of quadrilaterals where they are not fully or partially occluded by the instructor. The second step is to determine which ones of these un-occluded quadrilaterals has been updated by the instructor.

3.1. Step 1: finding un-occluded quadrilaterals regions:

To achieve this purpose, we first obtain a background model from a sequence of whiteboard training images (without writing and moving person). We then model the whiteboard background pixel color by on-line Gaussian mixture model, similar to those used in general background subtraction applications [5]. In order to improve the efficiency, we applied background subtraction to the sub-sampled image views at half of its original image dimension. The background model is updated during the real-time session.

We then applied the mathematical morphology to obtain one major connected foreground blob, shown in fig 5.b. The application of morphology operator and test for region connectivity excludes the whiteboard writing changes to be included as part of the foreground blob.

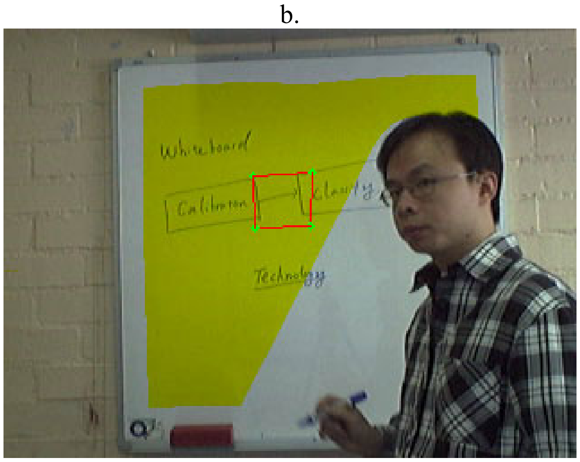
Take note that we set a tight constraint for un-occluded quadrilateral detection by finding a convex hull of the foreground blob's external contour, shown in fig 5.c. This is because the miss-detection of un-occluded quadrilateral is less prohibitive than false-detection, as the first case is likely to be corrected when the instructor moves away from the miss-detected quadrilateral.

3.2. Step 2: determine the updated quadrilaterals:

For the un-occluded quadrilateral set obtained in step 3.1, we then further classify them into the "updated" and "not-updated" ones. In each quadrilateral, we calculate the pixel difference between the current camera view and the corresponding pixels of the last-updated image using Normalized Cross-Correlation, similar to method described in [4]. Also, as a confirmation step, we use Canny Filter to detect the edges within the updated quadrilateral candidate, as whiteboard writing region contain significant amount of edges. This technique is also found in [6]. When an updated quadrilateral is found (fig 5.c), it is sent to the PTZ camera's scanning queue.

a.

b.



c.

Figure 5: (a) the static camera view, (b) the corresponding foreground blob (c), the yellow area is the whiteboard region excluding the enlarged convex hull of the foreground blob; the red lines indicates the quadrilateral to be scanned next.

4. Resultant Whiteboard Image

The final whiteboard images are displayed at a higher resolution (typically, we choose its width to be 1024 pixels) with its shape rectified to a rectangle, according to a pre-specified aspect ratio.

There are two transformations required to warp each quadrilateral obtained from the PTZ camera view to the viewing window. The first perspective transform T is:

$$\rho_1 \begin{bmatrix} static_x \\ static_y \\ 1 \end{bmatrix} = T \begin{bmatrix} ptz_x \\ ptz_y \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & 1 \end{bmatrix} \begin{bmatrix} ptz_x \\ ptz_y \\ 1 \end{bmatrix} \quad (3)$$

where (ptz_x, ptz_y) is the co-ordinate of one of the quadrilateral corners when PTZ camera is zoom close-up. ($static_x$, $static_y$) is its corresponding co-ordinate in the static camera view.

Since the system detects the entire corresponding quadrilateral corners in both static and PTZ camera during calibration, therefore, we can retrieve corner information to solve eight simultaneous equations to obtain matrix T. Similarly, we have another perspective transformation matrix S, which transforms the outermost corner of the whiteboard in the static camera view to four corners in the viewing window, i.e, *(0, 0)*, *(width, 0)*, *(width, height)*, *(0, height)*. Therefore, the pixels within each of the equilateral in the PTZ camera view can be transformed into its corresponding equilateral in the viewing window by perspective warping using matrix $S \bullet T$.

The captured whiteboard image is never uniformly white. This is due the shadows and camera noises, even the same region captured from the same camera at the different optical zoom level, may appear to have different colors. Similar to the efforts made in [3, 4], in our work, we enhance the resultant images, to make it to have a uniform white background.

During a training stage, we program the PTZ camera to zoom close-up to each of the quadrilaterals, using the parameters obtained by methods described in section 2.2. For each quadrilateral, we will collect 100 images. We will calculate a model image using their mean.

During a real-time session, when PTZ camera scans an image for a particular quadrilateral, the system compares, pixel by pixel, the difference between its corresponding model images. We keep the pixel color, when the difference is greater than a threshold T, and every other pixel will be assigned white background. The result of a rectified and enhanced image is shown in fig 6.

5. Limitation and Future works

In this paper, we present a prototype system, where we use a pair of static and PTZ camera to jointly capture and enhance the whiteboard image. Despite our initial success, one of the major drawbacks to our system is that the detection of un-occluded region in the static camera view, does not always guarantee the instructor to be absent from the corresponding quadrilateral in the PTZ camera view. This is because the two cameras differ in extrinsic parameter values (rotation and translation). Therefore, we usually place the two cameras close together and relatively far away from the whiteboard.

Our future experiment is to design a set of efficient methods which will use the calibrated static camera's parameters and the calibrated ranges of PTZ camera's intrinsic (at different zoom level) and extrinsic (at different pan-tilt levels), to track the 3D blobs of the instructor, and estimate the occluded region on the 2D whiteboard plane, for each of mechanical pan-tilt-zoom values that the PTZ camera will be set to.

References

[1] E. Saund, "Image Mosaicing and a Diagrammatic User Interface for an Office Whiteboard Scanner," Xerox Palo Alto Research Center 1999.

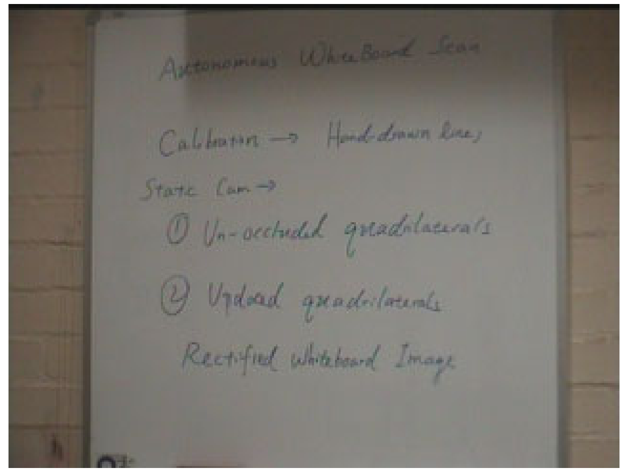
[2] G. A. Fink, M. Wienecke, and T. Plötz, "Experiments in video-based whiteboard reading," presented at Int. Workshop on Camera-Based Document Analysis and Recognition, Seoul, Korea, 2005.

[3] Z. Zhang and L. w. He, "Note-taking with a camera: whiteboard scanning and image enhancement," presented at IEEE International Conference on Acoustics, Speech, and Signal Processin (ICASSP '04), 2004.

[4] L. He and Z. Zhang, "Real-Time Whiteboard Capture and Processing Using a Video Camera for Remote Collaboration," *IEEE Transactions on Multimedia*, vol. 9, pp. 198 - 206, 2007.

[5] P. KaewTraKulPong and R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection," presented at 2nd European Workshop on Advanced Video Based Surveillance Systems, 2001.

[6] M. Wienecke, G. A. Fink, and G. Sagerer, "Towards Automatic Video-based Whiteboard Reading," presented at Seventh International Conference on Document Analysis and Recognition (ICDAR'03), 2003.



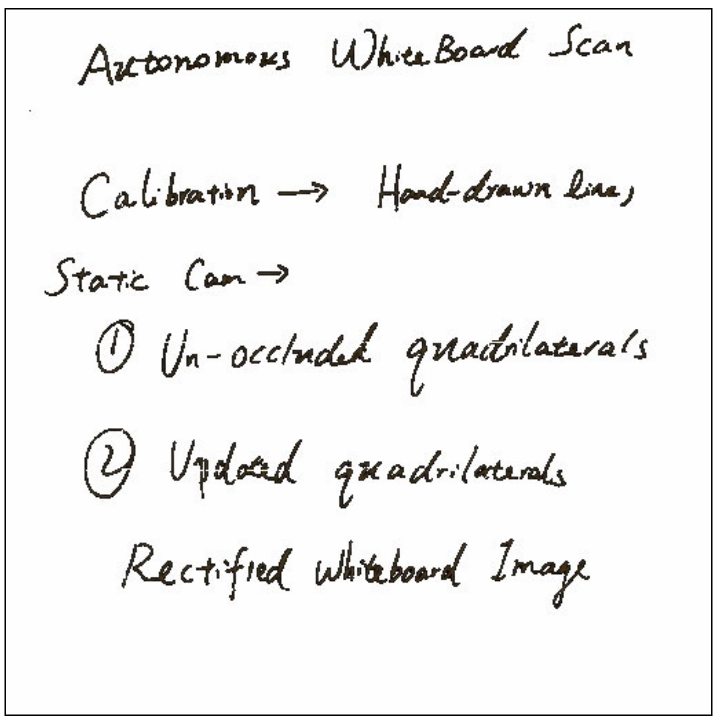


Figure 6: the rectified and enhanced whiteboard image