Real-time Video Surveillance for Large Scenes

Hanyu Liu
Technical Report

Abstract
Video surveillance provides security monitoring of target scenes. As the public safety is confronted of more and more neoteric challenges, numerous video surveillance cameras are deployed around the world for public safety and security, intelligent traffic management and homeland security. With the advances in both hardware and software, video surveillance systems have been innovated with many useful features, such as facial recognition, and integrated with networking technologies. However, constrained by optical technologies and costs, most cameras used in practice are designed to provide only limited angles of view, namely partially cover a scene, especially when it is large. Moreover, most of the cameras installed in current video surveillance are analog closed circuit television (CCTV) cameras that do not support IP capability for remote control, easy scaling in deployment and further mining of video information. This paper introduces a scalable multi-tier economic video surveillance solution with a unique feature—realtime video stitching. This system takes advantage of advance in networking technologies, cloud computing and computer vision. Preliminary studies have shown the merits of this solution.

I. INTRODUCTION
In the past years, a significant amount of effort and resources have been invested for homeland and public safety and the budget dedicated to sophisticated video surveillance systems are continuously increasing to cope with various safety and security challenges. In the United States, the Department of Homeland Security (DHS) issues billions of dollars every year to support local, state and federal agencies to maintain and upgrade video surveillance systems. In United Kingdom, the pioneer of public video surveillance, over 4.2 million CCTV cameras has been deployed. The operator behind the CCTV can even speak and hear through the loudspeaker and microphones mounted on the camera.

However, constrained by optical technologies, costs and networking bandwidth, most cameras are designed to provide only a limited resolution and angle, which limits the coverage of the monitored scene with only one camera. A widely used solution, as shown in Fig.1, to this limitation is to deploy multiple cameras with each covering a portion of the whole scene in a separated display or a viewing window in a large display. This solution is very intuitive, easy to deploy and have been widely used in many systems around the world. However, the separation of displays from different cameras monitoring the same scene provides offensive user experience and undermines the exploitation of useful information when navigating among those cameras. Furthermore, the traditional CCTV videos requires a special connection for the transmission of video streams, which can only be used within local areas. Comparably, cameras based on TCP/IP networking can delivery video over wired or wireless computer networks. Therefore, the captured real-time video can be easily shared with remote clients. This works well when only a limited number of cameras are involved. When we try to use many cameras to observe a large scene over a large network, a problem is introduced: it is difficult to synchronize all frames from different cameras due to different delays of these frames when they are transmitted over the packet-switching network.

In this paper, we present a novel framework, which combine the techniques in Computer Networking, Computer Vision and Human Computer Interaction to handle the problems mentioned above. In following section, we first describe the architecture of the system. Then, we discuss how to build a hierarchical real-time video stitching. Next, a novel working set oriented user interface is presented, which provides end users a more friendly and efficient way to accomplish their work with multiple IP cameras. Finally, some preliminary results and future works are discussed.

II. ARCHITECTURE
Our proposed distributed system is depicted in Fig. 2. Panasonic residential IP network cameras, BL-C210, are used to capture video, which support H.264, MPEG-4 and Motion JPEG compression formats. This camera also supports Real Time

Fig. 1: Existing video surveillance system, which have a number of CCTV cameras aiming to relevant area and TV wall to display the video from different sources.
Streaming Protocol (RTSP), which is a widely used multimedia communication control protocol over the Internet. As the prime task of RTSP is to maintain and control the session between the client and server, it relies, commonly, TCP to maintain an end-to-end connection. Therefore, all the components of the system, including cameras, clients and handset tablets, can be directly inter-connected to existing wired and wireless computer networks. In order to address the problem of different transmission delays for video streams from various cameras, an embedded device is installed near the cameras to perform video stitching, which need to be strictly synchronized, otherwise the difference of the transmission delay cannot be omitted when the network is very unstable. The video stitching, which is discussed in details in the following section, combines two frames at one time from both cameras into one and then streams out quietly. The computers and tablets used by the end users are installed with our novel user interface with video stitching functionality. The details of it would be also fully discussed later.

III. VIDEO STITCHING

It is critical and valuable for modern video surveillance systems to present a full coverage of a large scene while keeping necessary details. As shown in Fig. 1, the ill-designed TV wall is not a straight forward way to manage a number of cameras. Panoramic view is one of the possible way out. Immersive Media develops a solution from hardware to software. A number of cameras are mounted at one place with different orientation, then the videos from those cameras are mapped onto a sphere. So the output video could provide an immersive experience to the viewer. On the other hand, Panoramic Vision, from NEOVISION equips a standard camera with a non-planar mirror. The image from the camera is unwrapped by software and generates a panoramic image. However, those two systems are costly to video surveillance system, because they mainly focus on the immersion and need a large updating of hardware. In the following, we extends a traditional computer vision technique, image stitching, to real-time videos and encode the necessary information from the entire scene into the stitched unified video output.

A. Homography

Before we present video stitching, we first introduce Homography, a concept wildly used in Computer Vision. Homography can be viewed as an invertible transformation from one projective plane to another.

A 2D point in an image $X = (x, y)$ can be represented as a 3D vector $x = (x_1, x_2, x_3)$ where $x = \frac{x_1}{x_3}$ and $x = \frac{x_2}{x_3}$. This is referred as the homogeneous representation of a point and it lies on the projective plane $P^2$. A mapping from $P^2$ to $P^2$ is a projectivity if, and only if, there is a non-singular 3 by 3 matrix $H$, such that for any point in $P^2$, represented by vector $x$, it is true that its mapped point equals to $Hx$.

The most general 2-D transform is homography, an eight-parameter perspective transform and denoted by a $3 \times 3$ matrix $H$. The basic equation is as follows:

$$p_a = K_a H_{ba} K_b^{-1} p_b$$

where $P_a$ and $P_b$ are associating point pairs, in form of $p_a = [x_a, y_a, z_a]^T$, from image A and B respectively. $H_{ab}$ is a $3 \times 3$ homography matrix and $K$ is the intrinsic parameter matrix of a camera.

As mentioned above, a homography has 8 degree of freedom and 4 or more correspondences are needed to solve it. In our system, we use Scale Invariant Feature Transform (SIFT) to extract associating points and Random Sample Consensus (RANSAC) to eliminate outliers. SIFT first transforms both images into two collections of feature vectors, each of which is an invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric
Fig. 3: A collection of associating point pairs. Left shows the raw point pairs detected by SIFT; Right shows the result after RANSAC eliminated the outliers.

B. Parallel Stitching

As video can be decomposed into a sequence of frames, we extract frames from input video streams from IP cameras and perform image stitching at the frame speed of the video. In our system, a little modification is applied. The traditional method used in image stitching is shown in Figure 4a. One of its weaknesses is that the performance in each stitching step relies on its previous step. As a result, one step may block the whole pipeline no matter how much computation resource is still available.

To break the dependence among the steps in the serial stitching, expedite the stitching speed and fully utilize the computation power in modern computer architecture, we use a hybrid parallel stitching strategy to stitch a large number of images at acceptable Frame Per Second (FPS). First, we run SIFT and RANSAC on each pair of images. Then an adjacency matrix is built and each connection between two images is assigned the number of point pairs as a weight. With weight thresholds, each image is classified into one of a number of sets, in which the images are expected to have most associating point pairs and the frame with highest score is assigned as the master frame. Then, we compute the homography for all the other images in the same set that maps them onto the master frame. In practice, we take advantage of the OpenGL texture mapping and blending to achieve stitching and rendering, which is much faster than a looping implementation on the CPU.

C. Stitching on Embedded Device

In order to synchronize the videos from different camera via packet-switching networking, we modify our stitching program and deploy it on embedded devices. Our such devices have an AMD Geode Integrated Processor with 600 MHz clock speed.
Fig. 5: The upper three images are taken on a relevant area. Based on the strategy mentioned, the center image are selected as master frame and other two images are projected onto the mater frame. The lower image shows the stitched result, which presents the information of the whole area covered by three frames.

Considering the limited computational ability and RAM, only two cameras are connected to each device, when necessary. In addition, although SIFT and RANSAC are both computational intensive and need frequently memory access, they only need to be executed once in the initialization if the cameras are relatively stationary to each other. Our experiments showed the stitching on the embedded device can reach 15 FPS when two input videos both have 640 by 480 resolution. Then, the frames stitched from both cameras are streamed to the client with standard protocols such as RTSP, which provides an intermediate video stitching process transparent to the client.

D. Camera Calibration

In this subsection, we discuss a few things that can potentially improve the stitching performance.

As the camera’s inner parameters can be represented as a 3 × 4 matrix, which is called camera matrix, camera calibration is the process of estimating the real matrix. In our pinhole camera model, the projection from 3D real world to 2D pixel coordinates on a homogeneous point can be denoted as follow:

\[
zc \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[R \ T] \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}
\]

where \( A \) is the intrinsic parameters and \([ R \ T ]\) is the extrinsic parameters. In our system, only intrinsic parameters are required to be estimated. \( A \) can be denoted as

\[
A = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}
\]

where \( \alpha_x = f \cdot m_x \) and \( \alpha_y = f \cdot m_y \). In addition, \( f \) is the facial length, \( m \) is the scale factors, \( u \) and \( v \) are the offset between the pinhole center to the center of the image. We use a chess board, which is described in Zhang, to assist the process of calibration and restore an accurate representation of the real world.

There are also some other techniques to improve the realism of the stitched output video, such as vignetting correction and blending. However, considering the performance, including refresh rate, parallelism and reaction delay, we do not fully activate those functions in our system.

IV. WORKING SET ORIENTED USER INTERFACE

As each camera can only capture a small portion of the whole target scene, it is crucial to develop a novel user interface to assist a user to manage the cameras and accelerate the processing of those captured videos. The traditional grid TV wall, as in Fig. 1, lacks scalability and capability to customize. Once the system is set up, the layout and the order of the videos from the cameras can hardly be modified. We present a novel user interface which is built upon video entries. A video entry encapsulates basic video displaying and camera managing functions for each camera. An example of the interface is shown in Fig. 6. Each “entry” can be freely placed, resized and grouped on top of a flat panel, which can lead to a working set as in Fig. 7. The user can easily build up a highly customized and fully functional user interface to meet different requirements.
Fig. 6: Suppose 4 video from different cameras are opened and displayed by 4 video items, denoted as 4 black rectangles on left. If right mouse button is holden and draw a closed shape, all items enclosed are grouped as a working set. The operation applied on one item in a working set would be automatically forwarded to the other items in that working set.

Fig. 7: Suppose the two rectangles are two video items with different connected to different cameras. The contents displayed are represented in the top row and denoted with a number respectively. Once the user drag rectangle with “1” to the position overlapped with the other rectangle, 2, a new video item displaying stitched frame are initialized.

Fig. 8: A result based on two surveillance cameras, which cover a parking area.

If the user wants to stitch two videos, this can be achieved by simply dragging one “entry” to another with mouse or figures on touch screens. In addition, our user interface is also adaptive to different sizes of display. For large and high-resolution display, multiple “entries” can be added and stitched. On the other hand, for small displays or handset devices, one “entry” can be initialized and support basic video displaying function.

V. Conclusion and Future Works

In our preliminary experiments, our video stitching on the embedded device works at an average of 12 FPS output with two 640x480 video inputs. The user interface can easily boost a 30 FPS output with four 640x480 video inputs via RTSP over a typical IPv4 network. In addition, we conducted a small-scale pilot test and simulated both online course and parking area surveillance scenarios; all participants respond positively. In the next step, it would be interesting and meaningful to test the performance in practice and collect formal reposes from professional users. In addition, an automatic switch on frame blending and vignetting correction could be integrated into our system to improve the stitching performance when extra computing resources are available.