Active Multi-Camera Networks: From Rendering to Surveillance

Brian A. Stancil, Cha Zhang, *Member, IEEE*, and Tsuhan Chen, *Fellow, IEEE*

***Abstract*— Active multi-camera networks have a large and growing application base. On one end of the spectrum, active camera networks are being used to enhance the rendering or modeling of a single scene. Here, the many simultaneous views can be used to render a synthetic real time view even in a dynamically changing environment. The technical challenges include depth estimation within the scene, image correlation between multiple camera views, and manipulating the camera nodes in order to improve the rendered image. Active multi-camera networks are also being used to enhance surveillance applications for which a large area needs to be monitored. In these systems, a primary focus is tracking objects both within a single video feed as well as throughout a collection of video feeds. Active components are used to monitor larger areas and provide more continuous coverage of moving targets. Regardless of the application, real time processing constraints and bandwidth limitations constitute a significant problem with large networked camera systems.**

This paper presents an overview of these two highly researched application areas in the context of active multi-camera networks. For each, a breakdown of the typical approaches is presented along with a survey of real systems that implement them. We conclude with a brief discussion of the major research areas and future application potential combining the two technologies.

***Index Terms*—image-based rendering, multi-camera, Surveillance, active sensor networks**

I. Introduction

Image based rendering (IBR) techniques implemented with video camera networks have recently become extremely popular in the entertainment industry. Special effects such as the slow motion panorama used in the movie *The Matrix* and the EyeVision system developed by CBS and CMU [1] which produced the unique 360 degree stop action replays during the National Football League's championship game (Super Bowl XXXV) are both examples of IBR. In these systems, a network of cameras simultaneously captures an event such that it can be reproduced from an arbitrary viewpoint with little or no knowledge of the scene geometry. Compared with geometric models that dominate the traditional 3D rendering pipelines, images are easier to obtain, simpler to handle, and more realistic to render. Moreover, since image processing is such a widely studied research topic in literature, IBR has attracted many researchers from different communities including graphics, vision, and signal processing.

Video camera networks also constitute the heart of many surveillance systems today. Commercial off the shelf (COTS) video sensors are readily available and inexpensive, allowing for larger sensor networks. Commercial applications for networked video surveillance are extensive including the analysis of traffic flow, detection of accidents on the highway, compiling consumer demographics in shopping malls or amusement parks, and counting endangered species [2]. Military and security-based applications typically are associated with monitoring sensitive building perimeters or patrolling national borders. Video surveillance systems commonly provide live video feeds in which the system latency is low enough to allow for real-time monitoring decisions and control as well as a database capable of storing video and retrieving frames per camera for specific time intervals [3]. Historically, these systems often provided a common location (or control room) where all of the video data could be analyzed manually by a security officer or group of officers.

Each of these applications utilizes camera networks and therefore shares many of the same technical challenges. Video processing for rendering and surveillance purposes touches on many leading research areas in computer vision, pattern analysis, and artificial intelligence. Depending on the surveillance task, the sensor type, and the number of sensors in the network, the processing of the raw camera data can be quite a daunting task. Distributing the processing of the meta-level data to each of the sensing nodes in the network can dramatically reduce the bandwidth burden, allowing for larger and more capable networks. These independently operating nodes within a sensing network which provide a certain level of initial data filtering are often called "intelligent" or "smart" sensors.

As smart sensors evolve and embedded computing becomes more capable, modern systems are now utilizing controls which allow the sensor to become active. Stationary video sensor nodes might offer pan, tilt, or zoom capabilities. In a mobile surveillance system, the video sensors are actually

Manuscript received February 5th, 2008.

B. A. Stancil is with Applied Perception, a division of Foster-Miller, Pittsburgh, PA 16066 USA

C. Zhang, is with the Communication and Collaboration Systems Group at Microsoft Research, Redmond, WA 98052 USA

T. Chen is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA

attached to mobile platforms and can maneuver to a different spatial location or orientation (Figure 1). In either case, these active sensing nodes have the ability to reconfigure to obtain a better view of an area of interest, accumulate multiple views of an area of interest, or track a mobile target as it leaves the sensor field of view (FOV). In addition to providing better quality continuous video coverage of a target for surveillance purposes, actively manipulating the cameras in a network can also benefit IBR systems. In this context, the cameras can be moved so as to avoid occluded sections of the scene or improve the image quality of the synthesized image.

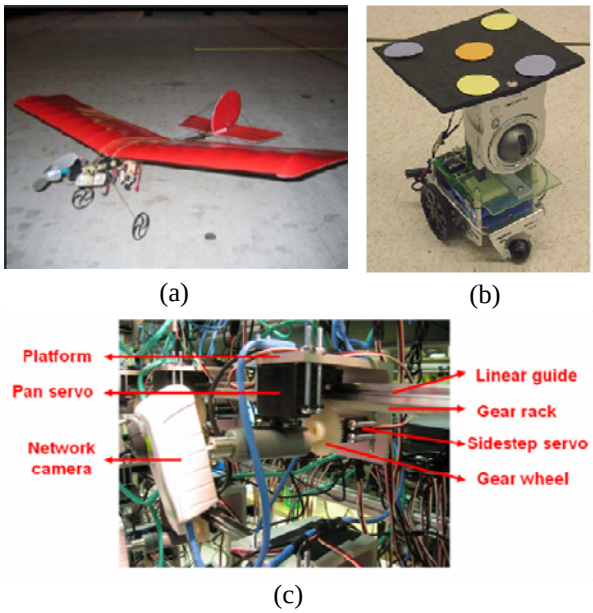


Figure 1: Examples of active camera platforms (a) Unmanned aerial camera [33] (b) mobile ground robotic platform [36] (c) rail mounted sliding camera [25]

This paper is divided as follows: First a discussion of active multi-camera IBR systems is presented including a brief survey of related work and a summary of the major challenges including geometry reconstruction (depth reconstruction), real-time rendering, and the use of active modules to improve rendering results. Section 3 treats the active multi-camera surveillance domain in the same way, discussing related work with respect to target detection/classification, motion tracking, data fusion, and planning and control optimization. We close by comparing and contrasting the two related applications and discussing potential future research areas including applications which make use of both technologies within a single system.

II. Active Multi-Camera Image-Based Rendering

Image based rendering has a rich history with many relevant surveys. Good reviews of the various image-based modeling techniques can be found in [4] and [5]. Utilizing a multi-camera array for IBR purposes can allow a system to render a higher quality image in real time due to the multiple simultaneous observations. Active camera components can be used to improve dynamically rendered scenes by maneuvering the cameras such that low quality areas of the scene are sampled at a higher resolution. The key tasks of an active multi-camera IBR system are discussed in the remainder of section 2. These include a brief review of scene depth and geometry reconstruction techniques, a survey of several multi-camera IBR systems and limitations with regards to real-time constraints, and finally a survey of approaches to active camera manipulation to improve IBR results.

A. Geometry Reconstruction

In IBR, when the number of captured images for a scene is limited, adding geometric information can significantly improve the rendering quality. In fact, there is a geometry image continuum which covers a wide range of IBR techniques, as is surveyed in [6]. In practice, accurate geometric models are often time consuming to generate. Many approaches in literature assume a known geometry, or acquire the geometry via manual assistance or a 3D scanner. Recently, there has been increasing interest in on-the-fly geometry reconstruction for IBR [7] [8] [9].

Depth from stereo is an attractive candidate for geometry reconstruction in real-time. Schirmacher et al. [7] built a 6-camera system which was composed of 3 stereo pairs which recovered depth on-the-fly. Within this system, each stereo pair used a dedicated computer for the depth reconstruction. Naemura et al. [10] constructed a camera array system consisting of 16 cameras. A single depth map was reconstructed from 9 of the 16 images using a stereo matching PCI board.

Matusik et al. [8] proposed image-based visual hull (IBVH), which rendered dynamic scenes in real-time from 4 cameras. IBVH is a clever algorithm which computes and shades the visual hull of the scene without having an explicit visual hull model. The computational cost is low thanks to an efficient pixel traversing scheme, which can be implemented with software only. Another similar work is the polyhedral visual hull [11], which computes an exact polyhedral representation of the visual hull directly from the silhouettes. Lok [12] and Li et al. [13] reconstructed the visual hull on modern graphics hardware with volumetric and image-based representations. One common issue of visual hull based rendering algorithms is that they cannot handle concave objects, which makes some close-up views of concave objects unsatisfactory.

An improvement over the IBVH approach is the image based photo hull (IBPH) [14]. IBPH utilizes the color information of the images to identify scene geometry, which results in more accurately reconstructed geometry. Visibility was considered in IBPH by intersecting the visual hull geometry with the projected line segment of each light ray in a particular view. Similar to IBVH, IBPH requires the scene objects' silhouettes to provide the initial geometric information; thus, it is not applicable to general scenes (where extracting the silhouettes could be difficult) or mobile cameras. Recently, Yang et al. [9] proposed a real-time consensus-based scene reconstruction method using commodity graphics hardware. Their algorithm utilized the Register Combiner for color consistency verification (CCV)

with a sum-of-square-difference (SSD) measure, and obtained a per-pixel depth map in real-time. Both concave and convex objects of general scenes could be rendered with their algorithm.

As modern computer graphics hardware becomes more programmable and powerful, the migration to hardware geometry reconstruction (HGR) algorithms is foreseeable. However, at the current stage, HGR still has many limitations. For example, the hardware specification may limit the maximum number of input images during the rendering [13] [9]. Algorithms that can be used on hardware are constrained. For instance, it is not easy to change the CCV in [9] from SSD to more robust metrics such as pixel correlations. When the input images have severe lens distortions, the distortions must be corrected using dedicated computers before the images are sent to the graphics hardware.

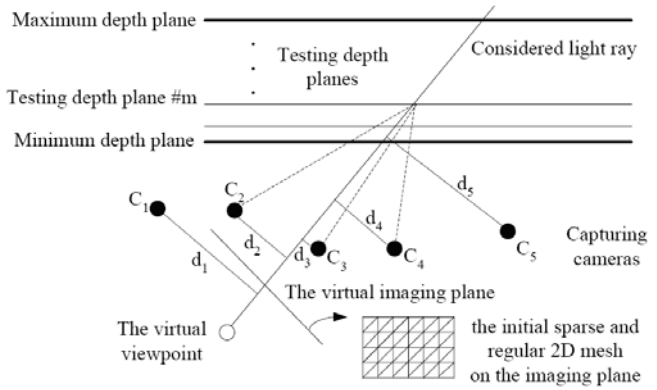


Figure 2: Locating the neighboring images for interpolation and depth reconstruction through plane sweeping. Here the cameras C2, C3, and C4 have the shortest distances and would be selected as the 3 closest images [15]

Zhang [15] reconstructs the scene depth of the light rays passing through the vertices of a 2D mesh using a plane sweeping method. Similar methods have been used in a number of previous algorithms [16] [17] [18], although they all reconstruct a dense depth map of the scene. As illustrated in Figure 2, the world space is divided into multiple testing depth planes. For each light ray, the scene is assumed to lie on a certain depth plane, and is projected into the nearby input images. If the assumed depth is correct, the colors are expected to be consistent among the projections. The plane sweeping method sweeps through all the testing depth planes, and obtains the scene depth as the one that gives the highest color consistency.

B. Real-time Image Based Rendering

Many existing IBR approaches are for static scenes. These approaches involve moving a camera around the scene and capturing many images. Novel views can then be synthesized from the captured images, with or without the scene geometry. In contrast, when the scene is dynamic, an array of cameras is needed. Recently there has been increasing interest in building such camera arrays for IBR. For instance, Matusik et al. [8] used 4 cameras for rendering using image-based visual hull (IBVH). Yang et al. [9] had a 5-camera system for real-time rendering with the help of modern graphics hardware; Schirmacher et al. [7] built a 6-camera system for on-the-fly processing of generalized Lumigraphs; Naemura et al. [10] constructed a system of 16 cameras for real-time rendering. These systems are illustrated in Figure 3.

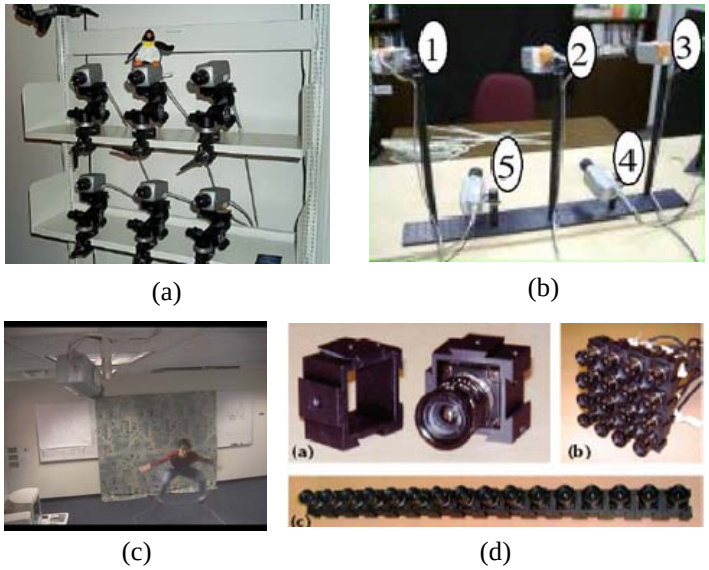


Figure 3: Examples of small camera arrays (a) Schirmacher et al. [7] (b) Yang et al. [9] (c) Matusik et al. [8] (d) Naemura et al. [10]

Several large arrays consisting of tens of cameras have also been built, such as the Stanford multi-camera array [19], the MIT distributed light field camera [18], the CMU 3D room [20], and the Reconfigurable Camera Array [15]. These four systems have 128, 64, 49, and 48 cameras respectively and are represented in Figure 4.

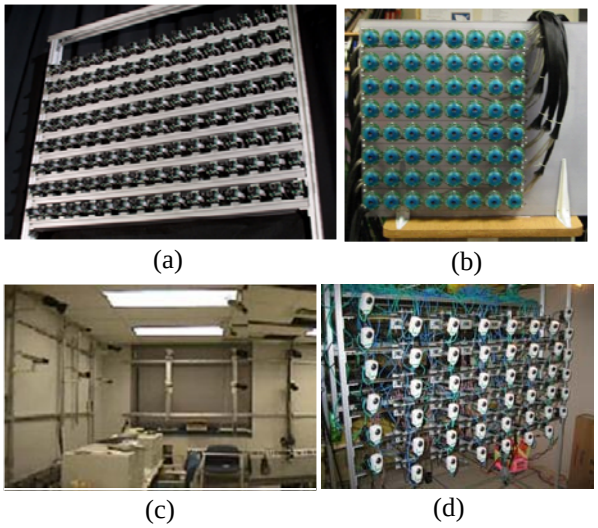


Figure 4: Large camera arrays (a) Stanford multi-camera array [19] (b) the MIT distributed light field camera [18] (c) CMU 3D room [20] (d) CMU Reconfigurable Camera Array [15]

In the above camera arrays, those with a small number of cameras can usually achieve real-time rendering [8] [9]. On-the-fly geometry reconstruction is widely adopted to compensate for the lack of cameras, and the viewpoint is often limited. Large camera arrays, despite their increased

viewpoint ranges, often have difficulty achieving satisfactory rendering speed due to the large amount of data to be handled. The Stanford system focused on grabbing synchronized video sequences onto hard drives. The CMU 3D room was able to generate good-quality novel views both spatially and temporarily [21]. It utilized the scene geometry reconstructed from a scene flow algorithm that took several minutes to run. While this is affordable for off-line processing, it cannot be used to render scenes on-the-fly. The MIT system did render live views at a high frame rate. However, this method assumes constant depth of the scene and can suffer from ghosting artifacts due to the lack of scene geometry. Such artifacts are unavoidable according to plenoptic sampling analysis [22] [23].

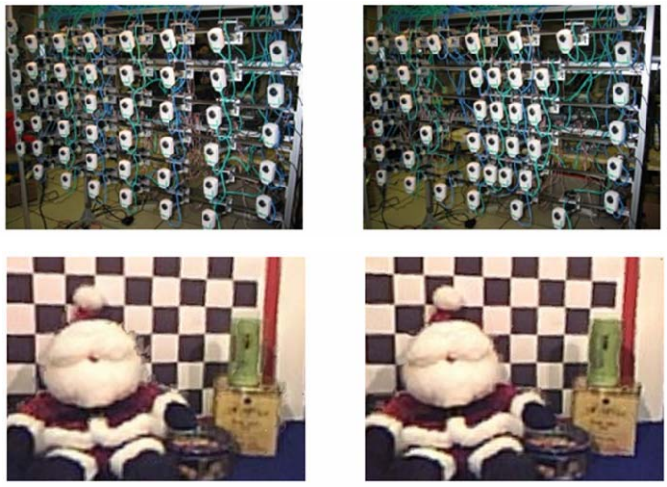
C. Planning and Control

Self-reconfiguration of the cameras is a form of non-uniform sampling (or adaptive capturing) of IBR scenes. In [24], Zhang and Chen proposed a general non-uniform sampling framework called the Position-Interval-Error (PIE) function. The PIE function led to two practical algorithms for capturing IBR scenes: progressive capturing (PCAP) and rearranged capturing (RCAP). PCAP captures the scene by progressively adding cameras at the places where the PIE values are maximal. RCAP, on the other hand, assumes that the overall number of cameras is fixed and tries to rearrange the cameras such that the rendering quality estimated through the PIE function is minimized. A small scale system was developed in [25] to demonstrate the PCAP approach. The work by Schirmacher et al. [26] shared similar ideas with PCAP.

One limitation concerning the above mentioned work is that the adaptive capturing process tries to minimize the rendering error everywhere as a whole. Therefore for a specific virtual viewpoint, the above work does not guarantee better rendering quality. Furthermore, since different viewpoints may require different camera configurations to achieve the best rendering quality, the final arrangement of the cameras is a tradeoff of all the possible virtual viewpoints, and the improvement over uniform sampling was not easy to show.

Zhang et al. [23] proposed the view-dependent non-uniform sampling of IBR scenes. Given a set of virtual views, the positions of the capturing cameras are rearranged to obtain the optimal rendering quality. The problem is formulated as a recursive weighted vector quantization problem, which can be solved efficiently. In that work it is assumed that all the capturing cameras can move freely on the camera plane. Such assumptions are very difficult to implement in practical systems.

In [15], Zhang et. al. uses error minimization techniques borrowed from stereo vision literature [27] [28] to maneuver a set of rail mounted cameras to improve a synthesized IBR image. Here the pixels within the synthesized view are back-projected onto the camera array plane. A CCV score is used which is calculated for each pixel to decide which cameras on the array should move. If two cameras have a pixel with a high CCV score which is back-projected into the space between them, they will be moved closer together so as to provide a denser sampling of that particular area. This process is shown to improve the overall quality of the synthesized view as seen in Figure 5. The major limitation of this system is that it is generally slow and is not suited for dynamically reconfiguring for a scene with motion.



(a) (b)

Figure 5: Scenes rendered by reconfiguring the camera array [15] (a) when cameras are evenly spaced (b) when cameras are self-reconfigured (note corrections near object edges)

III. Active Multi-Camera Surveillance

Visual surveillance has been a popular research topic for many years with recent focused efforts sponsored by DARPA in the United States, and sponsored by ESPRIT in Europe. Stimulated by the increased national interest, several good surveys and workshops have been conducted which explore the state-of-the-art in video surveillance [2] [29] [30]. In any active automated surveillance system, there are several key tasks that need to be performed. At the core, the system must be able to detect and classify targets within a video frame. Once the targets have been identified, they must be tracked as they move within the video sequence. In a multi-camera system, it is not only important to track targets within a single video feed, but also to track targets as they move in and out of each video camera FOV within the system. Lastly, automated active camera systems require a planning scheme in which they utilize the additional camera controls (pan/tilt/zoom/etc) to improve the system surveillance performance.

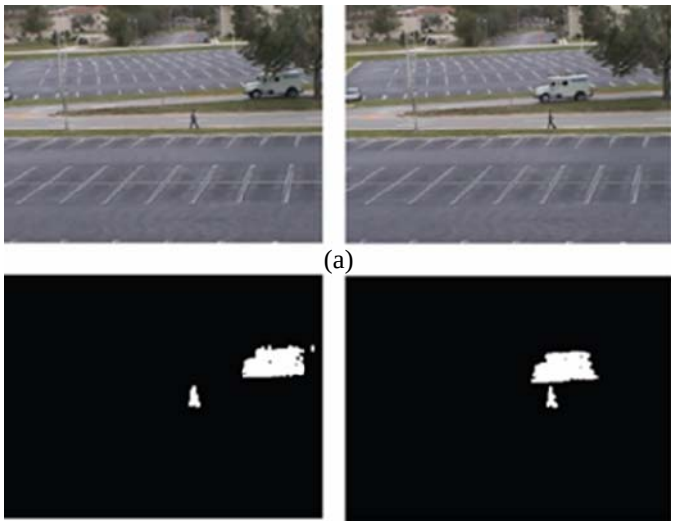
As with IBR systems, the utilization of a camera network raises similar challenges involving processing limitations, network limitations, data fusion between sensing nodes, and collaborative control algorithms. In the remainder of section 3, we will present a brief survey of how each of these challenges is addressed within the context of several multi-camera surveillance systems which are currently being used for both commercial and research purposes.

A. Single Sensor Surveillance Operations

Unlike many IBR applications, in which the rendered synthetic scene is typically the desired output, a critical component to most automated surveillance systems is the detection of moving targets for which classical approaches include temporal differencing, optical flow, and background

subtraction. In each sensor frame, targets must be identified. For video sensors, this involves identifying a subset of the pixels within a frame as a single distinguishable target. In many cases, it is useful to categorize the target into a specific object class such as a human or automobile.

The Knight surveillance system [31] developed at the University of Central Florida, performs object detection by using a background subtraction scheme which uses both color and gradient features to reduce sensitivity to daily variations in illumination. Subtraction is first performed at the pixel level to determine whether the pixel is considered to be part of the background or foreground. Foreground pixels are then segmented into groups within the image. These groups are then filtered depending on whether the pixels along the perimeter have high gradient changes based on the observation that interesting objects within images typically have well-defined boundaries. The resulting binary image is shown in Figure 6.



(a)

(b)

Figure 6: background subtraction within Knight (a) raw video (b) output of background subtraction [31]

The Knight system can classify people and vehicles using recurrent motion images (RMIs) of each target. These images have high values for pixels which undergo repeated motion, and lower values for areas of the target that remain constant. Targets with high valued RMIs can be considered human due to the repeated motion associated with walking or running; whereas automobiles tend to have a more consistent appearance as they move across the image. Target classification within the Video Surveillance and Monitoring (VSAM) system developed by Carnegie Mellon University and DARPA [32] uses two methods: a neural network classifier used to broadly categorize targets (such as human, human group, vehicle, or clutter) and a linear discriminant analysis variant used to distinguish between different types of automobiles. The latter method uses feature vectors that are generated from training examples to classify clusters of points that correspond to different automobile types.

Once targets have been detected within a single frame, an automated surveillance system needs to be able to correlate subsequent observations such that an object moving across the FOV, such as an automobile, is represented within the system as a single object rather than multiple discrete objects within each frame. This temporal correlation, or motion tracking, also enables a surveillance system to estimate motion parameters of each target such as velocity and heading.

Within the VSAM system [32], moving targets are detected using an adaptive background subtraction variation which utilizes both pixel-based and region-based processes. Here, individual pixels are observed for intensity fluctuations which are typical of a moving target as it passes through them. Groups of pixels which are similarly classified as moving or non-moving in space and time are divided up into layers within the image. Moving layers are associated between successive frames by performing image correlation matching. Estimates of the moving layer velocity and heading are collected by looking at the layer position over time. Additionally, multiple match hypothesis methods are used to decide when a layer exits a frame or overlaps with another layer within the FOV.

Of course, in an active camera surveillance system, the motion of the camera itself may need to be compensated for in order to extract an accurate measurement of other moving targets. The COCOA system [33] is a surveillance system implemented on small unmanned aerial vehicles (UAV). In order to detect motion, the video feeds first undergo ego-motion compensation. Here, a Harris corner detector is used to find features in the image, which are then matched to adjacent frames using RANSAC to determine the features with the best geometric correspondence. Once these features have been matched within adjacent frames, the images can be aligned to compensate for the UAV's motion as illustrated in Figure 7.

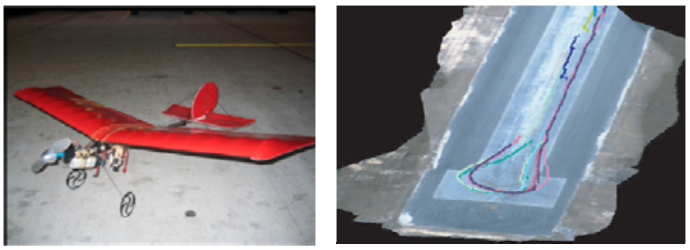


Figure 7: Unmanned Aerial Vehicle (left) and motion compensated image (right) [33]

In many cases [31][32][35], the computational and network load associated with detection, classification, and intra-camera tracking is mitigated by distributing these single sensor procedures to a network of smart sensors comprised of an imaging sensor and an accompanying processor board. In each case, target detection, classification and tracking is performed directly on the smart sensor which reduces the processing burden at the system layer. Additionally, network load is significantly reduced by only transmitting target level meta-data instead of raw image data. In the case where raw image data is desired for logging or direct observation, the pixels associated with specific targets can be transmitted at a higher frequency than the pixels associated with the background of

the image. The Kingston University Experimental Surveillance system (KUES) [35] performs this type of raw pixel filtering by only transmitting foreground pixels over the network.

B. Data Fusion (Tracking Between Cameras)

In a network of active surveillance cameras it is possible, and often desirable, to observe a target from more than one camera FOV in the same way that a multi-camera IBR system does. Historically in surveillance applications, the goal has not been to generate synthetic views of the target, but rather to maintain a single cohesive system level description of the target as it moves within the coverage area. This can happen asynchronously, if the target leaves the FOV of one camera and then enters the FOV of another camera, or simultaneously if the target is being observed by more than one camera at the same time. [34] refers to these redundant observation scenarios as spatio-temporal overlap, in which some representation of a specific object overlaps in space-time with another independent representation of the same object. Accurately fusing target observations together can allow for a less congested user interface, increased accuracy in target state estimation, and more efficient planning and control.

KUES [35] learns inter-camera geometry by measuring the correlation between tracked targets exiting one camera FOV and entering another camera FOV. Here, large amounts of training data are used to establish probabilistic models for entrance and exit zone correlations between cameras. Targets can be successfully tracked between cameras if they match the expected temporal delay and appearance histogram as predicted by the transition model. A multi-view tracking server is used to analyze the stored database data and generate tracking metadata including entry/exit/stop zones, camera topology, and major traffic routes. This metadata is used to support faster and more intuitive queries by a user.

For [31], each target generated by the object detection module is evaluated against previously observed targets in the system. During each system cycle, every pixel within each target region votes for a particular known target in the system. A target in the current frame is said to be associated with a previously known target in the system if a percentage of all the pixels vote for that previously known target. Linear velocity is measured for persistent targets in the system and is used to maintain an estimate of where the target is -- even when completely occluded by other objects in the scene. Targets are tracked between frames by learning the relationships between common entrance and exit points from each camera's FOV. A non-parametric Parzen window technique is used to estimate the space and time between each pair of cameras in the system. By measuring the position and velocity of a target moving out of view of one camera, the probability of that same target entering the FOV of an adjacent camera at a specific time can be approximated.

Collins et al. [32] uses a single central operator control unit process which maintains a database of known targets that have been observed by one or more of the networked cameras. Each target in the database is identified by its location in the global coordinate frame, its categorization, and its color. Redundant location data is fused using a propagation of 2D Gaussian covariances. At each discrete time point, a hypothesis for the location of each target is estimated using a linear velocity model. This hypothesis is then merged with measurements, taking into account the uncertainty covariance of each to produce a single combined "best guess" of the actual location of the object. The classification of a target at the system level is simply the most frequent classification given to that target by all observations, and the color of the target is set to be the most recent color observation of the target by a camera. Figure 8 shows two cameras operating cooperatively. A target is discovered within one camera's FOV and its trajectory is measured. A second camera is then able to pan to the correct location to visually track the target.

Figure 8: VSAM [32] tracking between cameras. (a) target identified in one camera (b) a camera with a better view is maneuvered to intercept the estimated target trajectory

The Robot-based Imaging Test-bed (RIT) [36] includes a network of wireless cameras mounted on small differential drive robotic platforms. The RIT contains a single localization server process which receives pose measurements of robotic cameras from each of the localization sources within the system. Two sources of robot localization exist: (a) an overhead camera which can identify the various robotic cameras in the system using colored tags mounted on top of each robot and (b) each robotic camera can identify its own location by comparing its current observation with a large set of stored database images.

Each of the robotic camera poses, as measured by the localization sources (either overhead camera or self-localization), contain a measure of the 2D pose (x,y,θ) as well as a standard deviation for each dimension $(\sigma_x,\sigma_y,\sigma_\theta)$. A partial extended Kalman filter (EKF) is used to provide a combined system-level pose for each robotic camera which uses a simple kinematics-based robot motion model and takes into account independent pose measurements as well as the statistical uncertainty for each.

In [34], correspondences between tracked objects observed by multiple cameras are used to generate a combined mosaic map from each of the camera video feeds. In order to do this, a planar surface area is assumed allowing registration of the tracked object paths within each camera view. Once the object trajectories are registered into a common coordinate frame, a mosaic of the video sequences can be generated by rotating the individual image sequences appropriately. In Figure 9, image (a) illustrates the combined mosaic, (b) shows the correspondence of the tracked objects between each of the

three cameras, and (c) shows the final correspondence between the objects.

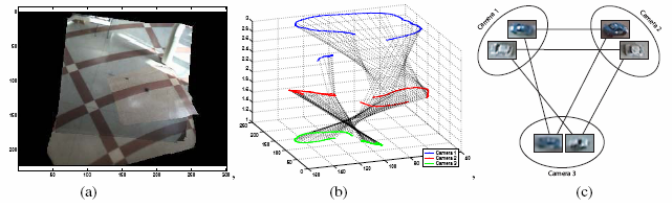


Figure 9: Generating a mosaic image by tracking targets using multiple cameras [34]

C. Planning and Control

This last step, the planning of camera actions and commanding motion or state changes, can occur at the sensor level or at the network level. In the former case, smart sensors can perform simple actions to improve the relevance of the metadata they provide to the network such as [37] and [32]. Examples such as panning to follow a moving target or zooming in on an area of interest to achieve higher resolution data would help to provide more continuous and detailed information to the system. Network level planning enables the system to collaborate between each camera within the network to best accomplish system-level goals.

A good example of network level planning is presented by Bramberger et. al [49] in which in which each smart camera in a multi-camera system stores a migration region consisting of the geometric region associated with the camera FOV, motion vectors representing target motion in the scene, and the next smart camera associated with each motion vector. The tracking task of a single target is passed from sensor to sensor along the predefined motion vectors as the target moves through the system thereby reducing the overall system processing so that other surveillance tasks can be performed.

Introducing mobility controls within a mobile surveillance system raises a number of new challenges. A video sensor on a mobile platform cannot rely on pre-calibrated localization. As the sensor is repositioned, the system must be able to actively recalibrate its location and orientation relative to the other sensing nodes in the system. Planning and control strategies need to be implemented such that the network of active sensors work together to service all of the necessary surveillance tasks optimally.

In [36], task arbitration is handled by allowing a user to define observation targets that represent an area which should be observed by one or more mobile camera nodes. Each observation target consists of a 2D position within the global coordinate frame and a user-defined priority. Higher priority translates to more robots congregating around the target yielding more simultaneous observations. Once the number of camera observers is decided for each target, specific cameras can be allocated to each target. Here, a greedy algorithm is used in which all of the robots claim their closest (linear distance) observation point. This allocation step can be continuously computed to adjust for dynamic changes in the environment or target allocation.

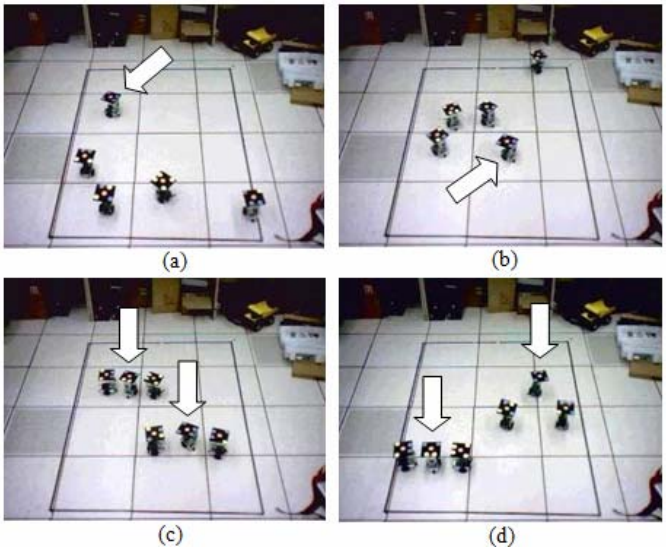


Figure 10: An example application in which target robot nodes (designated by white arrows) are observed by the other robots in the system [36]

This planning system is illustrated in Figure 10. In this example application, target nodes (designated by the white arrows) are observed by the other camera nodes in the system. In (a), the camera nodes are initially patrolling the four corners of an arena (designated by black tape). As the first target robot is introduced to the system, three of the cameras are allocated to the new target while one camera continues to patrol the parameter (b). When a second target is introduced into the system (c), the planning server divvies up the available resources such that two camera nodes are allocated to each target. The cameras continue to follow the target nodes as they move around the arena (d).

IV. Discussion

This paper has presented a survey of two application areas relevant to active networked multi-camera systems. IBR systems utilize networks of cameras to allow for the rendering of synthetic views of dynamically changing scenes. Here, the multiple simultaneously captured images are combined to reconstruct the virtual viewpoint in real time. Active camera components can be used to improve the synthetic image by maneuvering to view areas of the scene with poor rendering quality. Surveillance systems, on the other hand, have historically used camera networks to efficiently observe large areas simultaneously and in many cases, provide redundant views of a single target improving target feature estimation and reducing occlusion. Active components allow a surveillance system to track targets outside of a static FOV. In both cases, the use of an active multi-camera network raises key challenges associated with processing limitations, network bandwidth limitations, system level data fusion, and collaborative planning.

Each system deals with processor and network limitations differently based on the data necessary at the network level. For IBR, raw pixel data is needed from multiple cameras to form the synthetic images. Several techniques such as

intelligently selecting the most optimal cameras in the network, only transmitting the region of interest within each image [15] and assuming constant scene depth [16][17][18] help to mitigate these limitations. Even with these optimizations, there is still much to be done to further distribute IBR processes. Surveillance applications are typically more concerned with higher level meta-data (such as a targets position and trajectory) and therefore lend themselves to distributed processing via smart sensors. Similar tactics such as only transmitting video pixels associated with the target [35] and calculating target metadata such as position, color, or velocity [32] also help to alleviate network load. Network load requirements also impose hardware requirements in terms of the network medium (wireless/cabled) and the transmission protocol. Lin et. al. [48] provides tradeoff analysis of a multi-camera server-less system with regards to power and communication. Here, it is mentioned that camera placement is a driving design decision affecting the transmission power requirements and peak traffic over the network.

Data fusion is a straight forward, albeit difficult, challenge in IBR. Here, correlation algorithms are used to associate the pixels of each image such that an accurate estimation of a synthetic camera view can be generated. This can require a considerable amount of processing if the number of imaging sensors is large. CCV methods are used to perform this correlation and can be accelerated if knowledge of scene geometry is available. Hardware implementations of scene geometry calculations (HGR) may also help to offload the computational burden. However, expandability issues will need to be addressed if hardware solutions are to fuse a dynamically changing number of input images [13]. Surveillance applications need a combined representation of targets within the system. Several approaches including the a priori knowledge or online learning of camera FOV lines (or entrance and exit zones) can be used to associate targets traveling from one camera FOV to another. Probabilistic models such as Kalman filter variants [32][34] combined with trajectory modeling [36] are used when fusing simultaneous representations of a single target. In these cases, the standard challenge of accurately estimating the feature covariance matrix will determine the accuracy of the system level target description as well as the ability to accurately associate redundant measures of a single target.

As with data fusion, the goal for motion planning for active nodes is straightforward. Here, the system maneuvers the cameras to optimize the IBR calculation by cutting down on occlusion within the scene and obtaining redundant views which maximize the CCV rating of the synthetically generated view. In [15] this is accomplished by back-projecting CCV scores onto a camera plane. Cameras are then maneuvered to locations where the CCV score is sub-optimal to obtain a denser sampling in that location. This solution assumed that the cameras were mounted and maneuvered on a plane and does not adjust in real-time for dynamically changing scenes. Optimizing camera pose for IBR in full 3D in real time remains an open research topic. Surveillance applications, conversely, may want to “hand off” the task of tracking an object from one camera to another so that redundant observations are minimized and the monitored area is maximized [49]. Using smart sensors, single video feed planning operations such as intra-camera tracking can be offloaded to distributed microcontrollers which are bundled with each camera [32][36][37]. This simultaneously eases the processing burden of a centralized planning processor and cuts down on network utilization by control signals. At the network layer, planning and control for active surveillance networks is largely application driven. Sensors can be maneuvered to intercept motion trajectories of known targets [32] or allocated to track and follow an individual target specifically [36]. Active camera networks could also be used to maintain minimum communication links or optimize an ad-hoc network in terms of power or signal strength.

As these two application domains mature, it seems reasonable to speculate as to how they can be combined to form a more capable system. At this time, the quality of an IBR synthetic view generated by an active multi-camera system is limited in terms of mobility, processing capability, or both. By implementing an IBR system on a UAV or a mobile ground platform, it is possible to generate higher quality images. These new mobility degrees of freedom require more complicated algorithms for planning and control and increase the difficulty in determining accurate position and orientation estimates for the camera. Distributed target tracking as implemented on automated surveillance systems [31][32][35] could be used to select more relevant regions of interest for synthetic image generation. Furthermore, by using trajectory analysis and active camera nodes such as the ones presented in [32], an IBR system could seek to actively reconfigure itself to intercept and render a moving target as it moves through multiple camera FOVs. Surveillance systems could likewise benefit from IBR synthetic views. Novel views of an intruder could be rendered, providing additional intuition regarding the shape and depth of the scene. Additionally, IBR techniques could be used to allow many users to view a single scene simultaneously from different angles providing an effective way to evaluate multiple video feeds at one time.

This paper has served to highlight many of the application possibilities which exist for both IBR and surveillance within multi-camera networks. Given the collaborative benefits of each technology, it is likely that these systems will begin to merge as network bandwidth and processing limitations are alleviated by advances in hardware. As this occurs, the application specific architectural approaches of both IBR and surveillance to standard multi-camera network challenges will need to merge such that both tasks can be performed well.

REFERENCES

[1] CBS Broadcasting Inc. http://www.cbs.com. [Online]

[2] *Introduction to the Special Section on Video Surveillance.* R. T. Collins, A. J. Lipton, T. Kanade. 2000, IEEE, Transactions on Pattern Analysis and Machine Intelligence, pp. 745-746.

[3] *Scalable Video Requirements for Surveillance Applications.* A. May, J. The, P. Hobson, F. Ziliani, J. Reichel. 2004, IEE, pp. 17-20.

[4] *Image-based geometrically-correct photorealistic scene /object modeling (IBPhM): a review.* Zhang, Z. Y. Hong Kong : Asian Conference on Computer Vision (ACCV), 1998.

[5] *Image-Based Modeling and Rendering Techniques: A Survey.* Oliveira, M. RITA - Revista de Informatica Teorica e Aplicada, 2002, Vol. IX, pp. 37-66.

[6] *Survey of image-based representations and compression techniques.* H. Y. Shum, S. B. Kang, S. C. Chan.: IEEE Transaction on Circuit, SYstem on Video Technology, 2003. pp. 1020-1037.

[7] *On-the-fly processing of generalized lumigraphs.* H. Schirmacher, M. Li, H. P. Seidel: Eurographics, 2001.

[8] *Image-based Visual Hulls.* W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, L. McMillan: ACM Press /ACM SIGGRAPH, 2000. Proceedings of SIGGRAPH, Computer Graphics Proceedings. pp. 369-374.

[9] *Real-time consensus-based scene reconstruction using commodity graphics hardware.* R. Yang, G. Welch, G. Bishop: Pacific Graphics, 2002.

[10] *Real-time video-based modeling and rendering of 3d scenes.* T. Naemura, J. Tago, H. Harashima: IEEE Computer Graphics and ASpplications, 2002, pp. 66-73.

[11] *Polyhedral visual hulls for real-time rendering.* W. Matusik, C. Buehler, L. McMillan: Proceedings of Eurographics Workshop on Rendering, 2001.

[12] *Online model reconstruction for interactive visual environments.* Lok, B.: Proc. Symposium on Interactive 3D Graphics 2001, 2001.

[13] *Hardware accelerated visual hull reconstruction and rendering.* M. Li, M. Magnor, H. P. Seidel: Proc. of Graphics Interface, 2003.

[14] *Image-based photo hulls.* G. G. Slabaugh, R. W. Schafer, M. C. Hans. 2002.

[15] Zhang, C. *On Sampling of Image-based Rendering Data.* Pittsburgh : Dept. of Electrical and Computer Engineering, Carnegie Mellon University, 2004. PhD Thesis.

[16] *A space-sweep approach to true multi-image matching.* Collins, R. T.: Proc. of CVPR, 1996.

[17] *Photorealistic scene reconstruction by voxel coloring.* S. M. Seitz, C. R. Dyer: Proc. of CVPR, 1997.

[18] *A real-time distributed light field camera.* J. C. Yang, M. Everett, C. Buehler, L. McMillan: Eurographics Workshop on Rendering, 2002, pp. 1-10.

[19] *The light field video camera.* B. Wilburn, M. Smulski, H. H. K. Lee, M. Horowitz: SPIE Electronic Imaging, 2002. Proceedings of Media Processors.

[20] *The 3d room: Digitizing time-varying 3d events by synchronized multiple video streams.* T. Kanade, H. Saito, S. Vedula: Technical Report, CMU-RI-TR-98-34, 1998.

[21] Vedula, S. *Image Based Spatio-Temporal Modeling and View Interpolation of Dynamic Events*: Carnegie Mellon University, 2001. PhD Thesis.

[22] *Plenoptic Sampling.* J. X. Chai, S. C. Chan, H. Y. Shum, X. Tong: ACM Press /ACM SIGGRAPH, 2000. Proceedings of SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series. pp. 307-318.

[23] *Spectral analysis for sampling image-based rendering data.* C. Zhang, T. Chen: IEEE Transaction on Circuit, System on Video Technology, 2003, pp. 1038-1050.

[24] *Non-uniform sampling of image-based rendering data with the position-interval error (pie) function.* C. Zhang, T. Chen: Visual Communication and Image Processing, 2003.

[25] *A system for active image-based rendering.* C. Zhang, T. Chen: IEEE Int. Conf. on Multimedia and Expo (ICME), 2004.

[26] *Adaptive acquisition of lumigraphs from synthetic scenes.* H. Schirmacher, W. Heidrich, H. P. Seidel: EUROGRAPHICS, 1999.

[27] *A stereo matching algorithm with an adaptive window: Theory and experiment.* T. Kanade, M. Okutomi: IEEE Transaction on Pattern Analysis and Machine Intelligence, 1994, pp. 920-932.

[28] *Handling occlusions in dense multi-view stereo.* S. B. Kang, R. Szeliski, J. Chai: Proc. CVPR, 2001.

[29] *Intelligent distributed surveillance systems: a review.* M. Valera, S.A. Velastin. 2: Vision, Image and Signal Processing, IEE Proceedings, 2005, Vol. 152.

[30] *IEEE Workshops on Visual Surveillance* IEEE, 1998, 1999, 2000.

[31] *Automated Visual Surveillance in Realistic Scenarios.* M. Shah, O. Javed, K. Shafique: IEEE Computer Society, 2007, IEEE Computer Society, pp. 30-39.

[32] *Algorithms for Cooperative Multisensor Surveillance.* R. T. Collins, A. J. Lipton, H. Fujiyoshi, T. Kanade. 2001, Proceedings of the IEEE, pp. 1456-1477.

[33] *Detection and Tracking of Objects from Multiple Airborne Cameras.* M. Shah, A. Hakeem, A. Basharat. 2006, SPIE.

[34] *Object Tracking Across Multiple Independently Moving Airborne Cameras.* Y. Sheikh, M. Shah. 2005, IEEE International Conference on Computer Vision.

[35] *Wide Area Surveillance with a Multi Camera Network.* J. Black, T.J. Ellis, D. Makris. 2004, IEE, pp. 21-25.

[36] Stancil, B. *A Movile Robot Infrastructure for Vision-Based Planning and Control*: Carnegie Mellon University, 2007. MS Thesis.

[37] *Robust Detection and Tracking of Human Faces with an Active Camera.* D. Comaniciu, V. Ramesh: IEEE, 2000.

[38] *Into the Woods: Visual Surveillance of Noncooperative and Camouflaged Targets in Complex Outdoor Settings.* T. E. Boult, R. J. Micheals, X. Gao, M. Eckmann. 2001, Proceedings of the IEEE, pp. 1382-1402.

[39] *Sensor Fusion for Mobile Robot Dead-reckoning with a precision-calibrated Fiber Optic Gyroscope.* H. Chung, L. Ojeda, J. Borenstein: Robotics and Automation, 2001, IEEE International Conference on Robotics and Automation, pp. 3588 - 3593 .

[40] *Extended Kalman Filter based Mobile Robot Pose Tracking using Occupancy Grid Maps.* E. Ivanjko, I. Petrovic. Dubrovnik, Croatia : IEEE MELECON, 2004, IEEE Melecon, pp. 311-314.

[41] *Navigating Mobile Robots: Systems and Techniques.* J. Borenstein, B. Everett, and L. Feng. Wellesley : A. K. Peters, Ltd., 1996.

[42] Negenborn, R. *Robot Localization and Kalman Filters*: Institue of Information and Computing Sciences, Utrecht University, 2003. PhD Thesis.

[43] *A Multi-Sensor Surveillance System for Active-Vision Based Object Localization.* A. Bakhtari, M. Eskandari, M. D. Naish, B. Benhabib: IEEE, 2003.

[44] *Fast and accurate vision-based pattern detection and identification.* J. Bruce, M. Veloso. Pittsburgh : Dept. of Computer Science, Carnegie Mellon Univ.

[45] *Rendering with concentric mosaics.* H. Y. Shum, L. W. He: ACM Press / ACM SIGGRAPH, 1999. Proceedings of SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series. pp. 299-306.

[46] *A survey on image-based rendering - representation, sampling and compression.* C. Zhang, T. Chen: EURASIP Signal Processing: Image COmmunication, 2004, pp. 1-28.

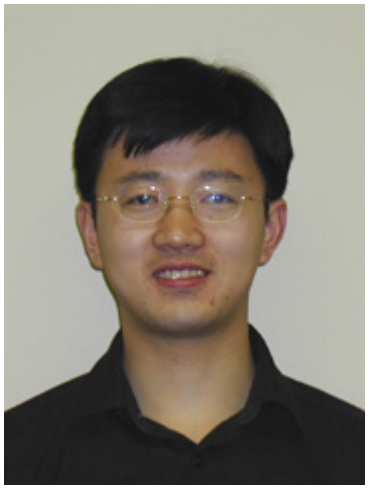
[47] *A survey of image-based rendering techniques.* Kang, S. B.: SPIE , 1999, VideoMetrics, Vol. 3641, pp. 2-16.

[48] *Design and Implementation of Ubiquitous Smart Cameras.* C. H. Lin, W. Wlf, A. Dixon, X. Koutsoukos, J. Sztipanovits: 2006, Proceedings of the IEEE International Conference on Sensor Networks, pp. 32-39.

[49] *Integrating Multi-Camera Tracking into a Dynamic Task Allocation System for Smart Cameras.* M. Bramberger, M. Quaritsch, T. Winkler, B. Rinner, H. Schwabach: 2005, IEEE, pp. 474-479.

Brian A. Stancil is currently a senior engineer at Applied Perception Inc, a division of Foster-Miller. He received his B.S. degree in computer science from Virginia Tech University (Blacksburg, VA) in 2002 and his M.S. degree in electrical and computer engineering from Carnegie Mellon University (Pittsburgh, PA) in 2007. Prior to working at Applied Perception, he worked at the National Robotics and Engineering Center at Carnegie Mellon University from 2002 to 2007. His current research interests focus on machine vision specifically with regards to object detection and classification.

He has worked on several autonomous land systems including the "Spinner" and "Crusher" platforms at the National Robotics and Engineering Center at Carnegie Mellon University under contract with DARPA, the Autonomous Navigation Systems program under the Future Combat Systems contract with the United States Army, and the Talon and LAGR platforms at Applied Perception in Pittsburgh, PA.



Dr. Cha Zhang is currently a Researcher in the Communication and Collaboration Systems Group at Microsoft Research, Redmond. He received the B.S. and M.S. degrees from Tsinghua University, Beijing, China in 1998 and 2000, respectively, both in Electronic Engineering, and the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University, in 2004. His current research focuses on applying various machine learning and computer vision techniques to multimedia applications, in particular, multimedia teleconferencing. He has worked on various multimedia related projects including sampling and compression of image-based rendering data, 3D model database retrieval and active learning for database annotation, peer-to-peer networking, etc.

Dr. Zhang has published more than 30 technical papers and holds numerous U.S. patents. He won the best paper award at ICME 2007. He co-authored a book titled Light Field Sampling, published by Morgan and Claypool in 2006.

Dr. Zhang has been actively involved in various professional activities. He was the Publicity Chair for International Packet Video Workshop in 2002, and the Program Co-Chair for the first Immersive Telecommunication Conference in 2007. He served as Technical Program Committee members for numerous conferences such as ACM Multimedia, CVPR, ICCV, ECCV, ICME, ICPR, ICWL, etc. He is an Associate Editor for Journal of Distance Education Technologies. He was a Guest Editor for the Advances in Multimedia Journal, special issue on Multimedia Immersive Technologies and Networking.

Dr. Tsuhan Chen has been with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, since October 1997, where he is currently a Professor and Associate Department Head. From August 1993 to October 1997, he worked at AT&T Bell Laboratories, Holmdel, New Jersey. He received the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, California, in 1990 and 1993, respectively. He received the B.S. degree in electrical engineering from the National Taiwan University in 1987.

Tsuhan served as the Editor-in-Chief for IEEE Transactions on Multimedia in 2002-2004. He also served in the Editorial Board of IEEE Signal Processing Magazine and as Associate Editor for IEEE Trans. on Circuits and Systems for Video Technology, IEEE Trans. on Image Processing, IEEE Trans. on Signal Processing, and IEEE Trans. on Multimedia. He co-edited a book titled Multimedia Systems, Standards, and Networks.

Tsuhan received the Charles Wilts Prize at the California Institute of Technology in 1993. He was a recipient of the National Science Foundation CAREER Award, from 2000 to 2003. He received the Benjamin Richard Teare Teaching Award in 2006, and the Eta Kappa Nu Award for Outstanding Faculty Teaching in 2007. He is elected to the Board of Governors, IEEE Signal Processing Society, 2007-2009. He is a member of the Phi Tau Phi Scholastic Honor Society. He is Fellow of IEEE, and a Distinguished Lecturer of the Signal Processing Society.