Chapter #X

A COMPARISON OF TECHNIQUES FOR CAMERA SELECTION AND HANDOFF IN A VIDEO NETWORK

YIMING LI AND BIR BHANU
University of California, Riverside

Abstract: Video networks are becoming increasingly important for solving many real-world problems. Multiple video sensors require collaboration when performing various tasks. One of the most basic tasks is the tracking of objects, which requires mechanisms to select a camera for a certain object and hand-off this object from one camera to another so as to accomplish seamless tracking. In this chapter, we provide a comprehensive comparison of current and emerging camera selection and hand-off techniques. We consider geometry, statistics, and game theory-based approaches and provide both theoretical and experimental comparison using centralized and distributed computational models. We provide simulation and experimental results using real data for various scenarios of a large number of cameras and objects for in-depth understanding of strengths and weaknesses of these techniques.

Key words: Utility-based Game Theoretic Approach; Co-occurrence to Occurrenc Ratio; Constraint Satisfaction Problem; Fuzzy-based

1. INTRODUCTION

The growing demand for security in airports, banks, shopping malls, homes, etc. leads to an increasing need for video surveillance, where camera networks play an important role. Significant applications of video network include object tracking, object recognition and object activities analysis from multiple cameras. The cameras in a network can perform various tasks in a collaborative manner. Multiple cameras enable us to have different views of the same object at the same time, such that we can choose one or some of them to monitor a given environment. However, since multiple cameras may be involved over long physical distances, we have to deal with the handoff problem. *Camera handoff* is the process of switching from the current camera to another one to follow an object seamlessly [1]. This has been an active area of research and many approaches have been proposed. Some camera networks require switches (video matrix) to help monitor the scenes in different cameras [2]. The control can be designed to switch among cameras intelligently. Both distributed and centralized systems are proposed. Some researchers provide hardware architecture design, some of which involve embedded smart cameras, while others focus on the software design for camera assignment. This chapter first gives a comprehensive review for the existing related works and then focuses on a systematic comparison of the techniques for camera selection and handoff. Detailed experimental comparisons are provided for four selected techniques.

This chapter is organized as follows: Section 2 gives a comprehensive background of the current and emerging approaches for camera selection and handoff. Comparisons tables are provided for a macroscopic view of the existing techniques. Section 3 focuses on the theoretical comparison and analysis of four key approaches. Experimental comparisons are provided in Section 4. Finally, the conclusions are drawn in Section 5.

2. RELATED WORK AND CONTRIBUTIONS

The research work in camera selection and handoff for a video network consisting of multiple cameras can be classified according to many different aspects, such as whether it is embedded /PC-based; distributed/centralized; calibration-needed/calibration-free; topology-based or topology-free; statistics-based/statistics-free, etc.

2.1 Comparison for Existing Works

Some researchers work on the design for embedded smart cameras, which, usually, consist of a video sensor, a DSP or an embedded chip and a communication module. In these systems, such as [3-7], since all the processing can be done locally, the design work is done in a distributed manner. There are also some PC-based approaches that consider the system in a distributed manner, such as [8-11]. Meanwhile, a lot of centralized systems are proposed as well, such as [12-16]. Some work, such as [16], requires the topology of the camera network while some are image-based and do not have requirements for any *priori* knowledge of the topology. As a result, calibration is needed for some systems, while some systems, such as [17-20] are calibration-free. Active cameras (pan/tilt/zoom cameras) are used

Table X-1. Merits of Various Characteristics Encountered in Distributed Video Sensor Networks

Properties	**Advantages**	**Disadvantages**
Distributed	Low bandwidth requirement; No time requirement for image decoding; Easy to increase the number of nodes.	Lack of global cooperation.
Centralized	Easy for cooperation among cameras; Hardware architecture is relatively simple compared with distributed systems.	Require more bandwidth; High computational requirements; May cause severe problem once the central server is down.
Embedded	Easy to be used in real-world distributed system; Low bandwidth.	Limited resources, such as memory, computing performance and power; Only simple algorithms have been used.
PC-based	Computation can be fast; No specific hardware design requirements, like for embedded chips or DSPs.	A bulky solution for many cameras.
Calibrated	Can help to know the topology of the camera network; A must for PTZ cameras.	Pre-processing is required; Calibration process may be time consuming.
Uncalibrated	No offline camera calibration is required.	Exact topology of cameras difficult.
Active cameras	Provide better view of objects; Can save the number of cameras by pa/tilt to cover larger monitoring range.	Camera calibration may be required, especially when zooming. Complex algorithms to account camera motions.
Static/Mobile cameras	Low cost, high for mobile; Easy to determine topology of the camera network; Relatively simpler algorithms as compared with those for active (and mobile) cameras.	More (statitc) cameras are needed to have a full coverage; Have no close-up if the object is not close to any cameras.

in some systems, such as [15, 16, 18], to obtain a better view of objects. However, to our knowledge, only a small amount of work has been done to propose a large-scale active camera network for video surveillance. More large-scale camera networks generally consist of static cameras.

Images in 3D are generated in some systems, such as [6]. However, in most approaches proposed for the camera selection and handoff, only 2D images are deployed. There are also other considerations, such as resource allocation [21], fusion of different types of sensors [22], etc. In Table X-1, we compare the advantages and disadvantages for some of the important issues discussed above.

Table X-2 lists sample approaches from the literature and their properties. It is to be noticed that, not all the distributed systems are realized in an embedded fashion. For instance, a distributed camera node can consisted of a camera and a PC as well, although the trend is to realize distributed systems via embedded chips. That is why we treat distributed

Table X-2. A Comparison for of Some Properties for Slected Approaches.

(Legends for the Table X-. HW-Hardware-wise; SW-Software-wise; E-Embedded; A-Active camera; D-Distributed; C-Calibration needed; RT-Real-time; RD-Real data; N_C-Number of cameras; N_P-Number of objects; T-Tracking algorithm used; O-Overlapping FOVs; Y-Yes; N-No; Y+ - Yes but not necessary)

Approaches	**HW**		**SW**		**Experiment Details**					
	E	***A***	***D***	***C***	***RT***	***RD***	N_C	N_P	***T***	***O***
Quaritsch *et al.* [4]	Y	N	Y	N	Y	Y	2	1	Camshift	N
Flech and Straβer [6]	Y	N	Y	N	Y	Y	1	1	Particle filter	Y
Park and *et al.* [7]	N	N	Y	N	N/A	N	20	N/A	N/A	Y
Morioka *et al.* [8]	N	N	Y	N	N/A	N	6	1	N/A	Y+
Morioka *et al.* [10]	N		Y	Y	Y	Y	3	3	Kalman filter	Y
Qureshi *et al.* [11]	N	Y	Y	Y	N	N	16	100	N/A	Y+
Kattnaker *et al.* [13]	N	N	N	N	Y	Y	4	2	Bayesian	N
verts and *et al.* [15]	N	Y	N	Y	Y	Y	1	1	Histogram based	N
Li and Bhanu [17]	N	N	N	N	Y	Y	3	2	Camshift	Y+
Javed and *et al.* [18]	N	N	N	N	Y	Y	2	2	N/A	Y
Jo and Han [20]	N	N	N	N	Y	Y	2	N/A	Manual	Y
Gupta et al. [23]	N	N	N	N	Y	Y	15	5	M2Tracker	Y
Song et al. [24]	N	N	Y	N	N	Y	7	9	Particle filter	N
Song et al. [25]	N	Y	Y	N	N	N	14	N/A	N/A	Y

systems and embedded systems separately in Table X-1. In Table X-2, some approaches are tested using real data while some provide only the simulation results. There is no guarantee that the systems, which are experimented using synthetic data, can still work satisfactorily and realize real-time processing when using real data. So, the real-time property is left blank for those approaches whose experiments use simulated data. Similarly, most of the experiments are done for a small-scale camera network. The performance of the same systems for a large-scale camera network still needs to be

2.2 Our Contributions

The contributions of this chapter are:

- A comprehensive comparison of recent work is provided for camera selection and handoff. Four key approaches are compared both theoretically and experimentally.
- Results with real data and simulations in various scenarios are provided for an in-depth understanding of the advantages and weaknesses of the key approaches. The focus of comparison is solely on multi-object tracking using non-active multi-cameras in an uncalibrated system. The comparison considers software and algorithm related issues. Resource allocation, communication errors and hardware design are not considered.

3. THEORETICAL COMPARISON

We selected four approaches [9, 11, 17 and 20] for comparison. They are chosen as typical approaches because these approaches cover both distributed systems [9, 11] and centralized systems [17, 20]. Although none of these approaches needs camera calibration, some of them do a geometry correspondence [20] while some do not [9, 11, 17]. Approaches such as [11, 17] provide a more systematic approach to camera selection and handoff. This section focuses on the comparison of theoretical ideas for while experimental comparison is provided in the next section. In this section, we first describe the key ideas of these approaches. Analysis of the advantages and disadvantages are provided in Table X-3.

Table X-3. Relative Merits and Shortcomings of the Selected Approaches

Approaches	**Pros**	**Cons**
Utility-based Game Theoretic Approach [17]	Provides a mathematical framework; Can deal with the cooperation and competition among cameras; Can perform camera selection based on user-supplied criteria.	Communication among cameras is not involved, can be extended for distributed computation; The local utility has to be designed that will align with the global utility in a potential game.
Co-occurrence to Occurrence Ratio Approach [20]	Intuitive efficient approach; Acceptable results when there are few occlusions and few cameras and objects.	Time consuming; When correspondence fails or occlusion happens, there is ambiguity; Becomes complicated when # of camera nodes/objects increases; FOVs have to overlap.
Constraint Satisfaction Problem Approach [11]	Provides a distributed system design; Camera nodes can cooperate by forming coalition groups; Conflicts among cameras are solved by the CSP.	The backtracking approach is time consuming for solving the constraint satisfaction problem; Only simple constraints are provided; Only simulation (no real video) results are provided.
Fuzzy-based Approach [8]	Distributed approach; Camera state transition and handoff rules are both intuitive.	Only simulation results are provided; Tracking has to be accurate; Not robust when occlusion happens; No guarantee for convergence in a large-scale network.

3.1 Descriptions of the Key Ideas of Selected Approaches

3.1.1 The Utility-based Game Theoretic Approach

This is the most systematic approach among the selected ones. It views the camera selection and handoff problem in a game theoretic manner. There is the trend to consider the camera assignment problem as a cooperative multi-agent problem. The merit of [17] is that the authors come up with a complete mathematical mapping of the problem to a classical vehicle-target problem in game theory by viewing the cameras that can “see” an object as the multiple players in a game. The problem formulation considers both cooperation and competition among cameras for tracking an object, which demonstrates the main advantage of applying game theory.

Camera utility, person utility, and the global utility are calculated:

- Global utility: $\mathrm{U_g}(a) = \sum_{C_j \in C} U_{C_j}(a)$ (1)
- Person utility for P_i: $U_{P_i}(a) = U_g(a_i, a_{-i}) - U_g(C_0, a_{-i})$ (2)
- Camera utility for C_j: $U_{C_j}(a) = \sum_{i=1}^{n_P} \sum_{l=1}^{N_{Crt}} Crt_{il}$ (3)

$a = (a_i, a_{-i})$ is the camera assignment result. a_i stands for the camera used to track person P_i , while a_{-i} stands for the camera assignment for all the other persons other than P_i. The person utility implies the marginal contribution of camera a_i to the global utility. Crt_{sl} are the user-supplied criteria. It is shown that the design of the utility functions as above makes it a potential game. The final assignment result is given in the form of a mixed strategy:

$$p_i^l(k) = \frac{e^{\frac{1}{\tau}\overline{U}_{P_i}^l(k)}}{e^{\frac{1}{\tau}\overline{U}_{P_i}^1(k)} + \cdots + e^{\frac{1}{\tau}\overline{U}_{P_i}^{n_C}(k)}} \qquad (4)$$

where

$$\overline{U}_{P_i}^l(k+1) = \begin{cases} \overline{U}_{P_i}^l(k) + \frac{1}{p_i^l(k)}(U_{P_i}\big(a(k)\big) - \overline{U}_{P_i}^l(k), a_i(k) = A_i^l \\ \overline{U}_{P_i}^l(k) & , otherwise \end{cases} \qquad (5)$$

is the predicted person utility in the $(k+1)^{th}$ iteration step. Due to the limited space of this chapter, for more detailed explanations, please refer to [17].

3.1.2 The Co-occurrence to Occurrence Ratio (COR) Approach

This approach decides whether two points are in correspondence with each other by calculating the co-occurrence to occurrence ratio (COR). If the COR is higher than some predefined threshold, then the two points are decided to be in correspondence with each other. When one point is getting close to the edge of the field of view (FOV) of one camera, the system will hand-off to another camera that has its corresponding point.

The COR is defined as

$$R(x, x') = \frac{p(x,,x')}{p(x)} \qquad (6)$$

where

$$p(x) = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N_t} K_2(x - x_t^i) \qquad (7)$$

is the mean probability that a moving object appears at x, i.e. the occurrence at x. K_2 is claimed to be circular Gaussian kernel. Similarly,

$$p(x,x') = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N_t} K_2(x - x_t^i)\sum_{i=1}^{N_t'} K_2(x' - {x'}_t^i) \quad (8)$$

is the co-occurrence at x in one camera and x' in another camera.

It is intuitive that if two points x and x' are in correspondence, i.e. the same point in the views of different cameras, then the calculated COR should be 1 ideally. On the contrary, if the x and x' are completely independent on each other, i.e. two distinctive points, then $p(x,x') = p(x)p(\mathrm{x}')$, which leads the COR $R(x,x')$ to be $p(x')$. These are the two extreme cases. If we chose some threshold θ_r such that$p(x') < \theta_r < 1$, then by comparing with θ_r, the correspondence of two points in two camera views can be determined. Another threshold θ_0 is needed to be compared with *p(x)* to decide whether a point is detected in a camera. Thus, camera handoff can be taken care of by calculating the correspondence of pairs of points in the views of different cameras and performed when necessary.

3.1.3 The Constraint Satisfaction Problem (CSP) Approach

The approach discussed in [11] focuses on the system design. Unlike the previous two centralized systems, this system is designed to be distributed by deploying the local visual routines (LVRs). Camera controllers are modeled as a finite state machine with the Idle state, the ComputingRelevance state and the PerformingTest state. The cameras cooperate with each other by forming coalition groups, which is achieved by involving leader nodes and the auction/bidding mechanism for recruiting new nodes. When multiple cameras nodes are available for joining the group, a conflict resolution mechanism is realized by solving the constraint satisfaction problem.

Three elements of a CSP are a set of variables $\{v_1,\ v_2, \ldots, v_k\}$, the domain of each v_i Dom$[v_i]$ and a set of constraints $\{C_1,\ C_2, \ldots, C_m\}$. The authors apply backtracking to search among all the possible solutions and rank them according to the relevance to solve the CSP. *BestSolv*, which is based on the quality of the partial solution, is compared with the *Allsolv*, which is an exhaustive manner.

3.1.4 The Fuzzy-based Approach

This is another decentralized approach. Each candidate camera has two states for the object that is in its FOV: the **non-selected** state and the **selected** state for tracking. Then, camera handoff is done based on the camera's previous state $\boldsymbol{S_i}$ and the tracking level state $\boldsymbol{SS_i}$, which is defined by estimating the position measurement error in the monitoring area. The two states for the tracking level are: **unacceptable**, meaning that the object is too far away and acceptable, meaning that the object is within the FOV and the quality is **acceptable.**

The block diagram for camera state transition and the fuzzy rule for camera handoff are given in Fig. X-1 [9] and Fig. X-2 [9], respectively.

4. EXPERIMENTAL RESULTS

In this section, we perform experiments for the above four approaches in different cases. Although some of the approaches [9, 11] do not have results with real data, in this chapter, both indoor and outdoor experiments with real data are carried out for all the approaches. For convenience of comparison among different approaches, no cameras are actively controlled.

4.1 Data

The experiments are done using commercially available AXIS 215 cameras. Three experiments are carried out with an increase in complexity. *Case 1:* 2 cameras 3 persons, indoor. *Case 2*: 3 cameras 5 persons, indoor. *Case 3:* 4 cameras 6 persons, outdoor. The frames are dropped whenever the image information is lost during the transmission. The indoor experiments use cable-connected cameras, with a frame rate of 30 fps. However, for the outdoor experiment, the network is wireless. Due to the low quality of the images, the frame rate is only 10-15 fps on average. The images are 60% compressed for the outdoor experiment to save bandwidth. Images are 4CIF,

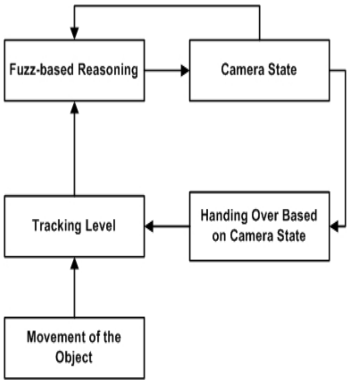

Figure X-1. Diagram for camera state transition.

(1) If S_i = ***Selected*** And SS_i = ***Acceptable*** Then S_i = ***Selected***

(2) If S_i = ***Non-selected*** And SS_i = ***Unacceptable*** Then S_i = ***Non-selected***

(3) If S_i = ***Selected*** And SS_i = ***Non-selected*** And SS_k = ***Unacceptable*** Then S_i = ***Selected***, $\forall k \in [1, N], k \neq i$, where N is the number of camera candidates

(4) If S_i = ***Non-selected*** And SS_i = *Acceptable* And S_k = ***Non-selected*** And SS_k = ***Unacceptable*** Then S_i = ***Selected***, $\forall k \in [1, N], k \neq i$

(5) If S_i = ***Non-selected*** And SS_i = ***Acceptable*** And S_k = ***Selected*** And SS_k = ***Acceptable*** Then S_i = ***Non-selected***, $\exists k \in [1, N], k \neq i$

(6) If S_i = ***Selected*** And SS_i = ***Unacceptable*** And S_k = ***Non-selected*** And SS_k = ***Acceptable*** Then S_i = ***Non-selected***, $\exists k \in [1, N], k \neq i$

(7) If S_i = ***Non-selected*** And SS_i = ***Acceptable*** And S_k = ***Selected*** And SS_k = ***Unacceptable*** Then S_i = ***Selected***, $\exists k \in [1, N], k \neq i$

Figure X-2. Fuzzy-based reasoning rules.

which 704×480. They are overlapped randomly in our experiments, which is not required by some of the approaches but required by some others.

4.2 Tracking

None of the approaches discussed here depends on any particular tracker. Basically, ideal tracking can be assumed for comparing the camera selection and handoff mechanisms.

Trackings in all the experiments are initialized manually at the very beginning and then done with color-based particle filter. The dynamic model used is random walk. Measurement space is 2 dimensional: hue and saturation values of a pixel. The sample number used for each object to be tracked is 200 for indoor experiments and 500 for outdoor experiments. Tracking can be done in real-time by implementing the OpenCV structure CvConDensation and the corresponding OpenCV functions. Matches for objects are done by calculating the correlation of the hue values using cvComparehist Minor occlusion is recoverable within a very short time. Tracking may fail when severe occlusion takes place or the case that an object is not in the scene for too long and then re-enters. Theoretically, this can be solved by spreading more particles. However, more particles is computationally expensive. Thus, we just re-initialize the tracking process.

4.3 Parameters

We first define the following properties of our system:

- A person P_i can be in the FOV of more than one camera. The available cameras for P_i belong to the set A_i.
- A person can only be assigned to one camera. The assigned camera for P_i is named as a_i.
- Each camera can be used for tracking multiple persons.

1) The Utility-based Game Theoretic approach: The utility functions are kept exactly the same as they are in [17]. The criteria used for calculating the cameras are the combined criterion mentioned in [17], i.e. a weighted sum of the other three criteria. For instance, the criterion for P_i is calculated as:

$$Crt_i = 0.2Crt_{i1} + 0.1Crt_{i2} + 0.7Crt_{i3} \quad (9)$$

$a)$ Crt_{i1}: *The size of the person*. It is measured by

$$r = \frac{\#\ of\ pixels\ inside\ the\ bounding\ box}{\#\ of\ pixels\ in\ the\ image\ plane} \quad (10)$$

Assume that λ is the threshold for best observation, i.e. when $r = \lambda$ this criterion reaches its peak value, then

$$Crt_{i1} = \begin{cases} \frac{1}{\lambda} r, \ when\, r < \lambda \\ \frac{1-r}{1-\lambda}, \quad when\, r \geq \lambda \end{cases} \tag{11}$$

b) Crt_{i2}: *The position of the person in the FOV of a camera.* It is measured by the Euclidean distance that a person is away from the center of the image plane

$$Crt_{i2} = \frac{\sqrt{(x-x_c)^2+(y-y_c)^2}}{\frac{1}{2}\sqrt{{x_c}^2+{y_c}^2}} \tag{12}$$

where *(x,y)* is the current position of the person and (x_c, y_c) is the center of the image plane.

c) Crt_{i3}: *The view of the person.* It is measured by

$$R = \frac{\#\, of\ pixels\ on\ the\ face}{\#\, of\ pixels\ on\ the\ entire\ body} \tag{13}$$

We assume that the threshold for best frontal view is , i.e. when R = ξ the view of the person is the best, where

$$RCrt_{i3} = \begin{cases} \frac{1}{\xi} r, \ when\, R < \xi \\ \frac{1-R}{1-\xi}, \quad when\, R \geq \xi \end{cases} \tag{14}$$

2) The COR Approach: The COR approach in [20] has been applied to two cameras only. We generalize this approach to the cases with more cameras by comparing the accumulated COR in the FOVs of multiple cameras. We randomly select 100 points on the detected person, train the system for 10 frames to construct the correspondence for these 100 points, calculate the cumulative CORs in the FOVs of different cameras and select the one with the highest value for hand-off.

3) The CSP Approach: According to the assumption made earlier, we allow one camera to track multiple persons but one person can only be tracked by one camera. So, for each camera C_j, we let all those persons that can be seen by this camera form a group g_j. For instance, if, in our case, the camera C_j can see person P_1 and P_2, then the domain of g_j , noted as Dom[g_j], is *{*$\{P_1\}$, *{*P_2*}*, $\{P_1,\ P_2\}$*}*.The constraint is set to be $d_i \cap d_j = \{\emptyset\}, for\ i \neq j$, where $d_i \in b_i \cup \emptyset$ is the camera assigned to track person P_i. , where b_i and b_j belong to Dom[g_j] and $i \neq j$. By doing so, we mean that the persons to be tracked are assigned to different cameras.

4) Fuzzy-based Approach: We apply the same fuzzy reasoning rule as the one in Figure 2, which is given in [9]. The tracking level state is decided by the Criterion 2, i.e. Crt_{i2}, which is used for the utility-based game theoretic approach.

4.4 Experimental Results and Analysis

Due to limited space, only those frames with camera handoffs are shown (actually, only some typical handoffs, since the video is long and there are too many handoffs.). These camera handoffs for case 1-3 are shown in Fig. X-3 to Fig. X-5 respectively. Since no topology of the camera network is given, tracking is actually performed by every camera all the time. However, for easy observation, we only draw the bounding box for an object in the image of the camera which is selected to track this object. Case 1 and Case 2 are simple in the sense that there are fewer cameras and objects and the frame rate is high enough to make the objects trajectories continuous. So, we only show some typical frames for these cases and give more handoff examples in Case 3, which is more complicated. We show some typical

Figure X-3. Selective camera handoff frames for the four approaches (Case 1).

Figure X-4. Selective camera handoff frames for the four approaches (Case 2).

Figure X-5. Selective camera handoff frames for the four approaches in case 3.

handoffs for Case 1 and Case 3, while for Case 2, we show the same frames for the four approaches to see the differences caused by different

It is clear that the utility-based game theoretic approach considers more criteria when performing the camera selection. Camera handoffs take place whenever a better camera is found based on the user-supplied criterion in this case. So, cameras that can see persons' frontal views, which has the highest weight in Crt_i, are more preferred most of the time. The other three approaches have similar results in the sense that they all consider handoff based on the position of the objects. Ideally, handoffs should take place near the FOV boundaries most of the time. Different results are caused by different iterative methods to get the solutions. The design for new constraints and tracking levels are non-trivial. On the contrary, if we just want to consider camera handoffs when a person is leaving and entering the FOV of a camera by using the utility-based game theoretic approach, we can achieve this by just apply the Criterion 2 in [17]. In [17], this is compared with the results using the combined criterion. Based on the error definition, the combined criterion produces much better results. In this sense, the game theoretic approach is more flexible to perform camera handoffs based on different criteria. The modification of a criterion will have no influence on the decision making mechanism. Fig. X-3 shows the camera handoff results for a very simple case. All the four approaches achieve similar results, although the utility-based game theoretic approach prefers frontal view.

As the scenario being more complex, the COR approach and the fuzzy-based approach have less satisfactory results. The CSP approach needs relatively long time for computing the solutions when the camera network is growing larger, as what is shown in Fig. X-6. Error rates for different approaches in the each case are given in Table X-4.

Table X-4. Error Rates of the Selected Approaches.

	Utility -based	**COR**	**CSP**	**Fuzzy-based**
Case 1	3.86%	4.23%	3.92%	4.64%
Case 2	4.98%	10.01%	6.33%	7.11%
Case 3	7.89%	45.67%	12.96%	21.33%

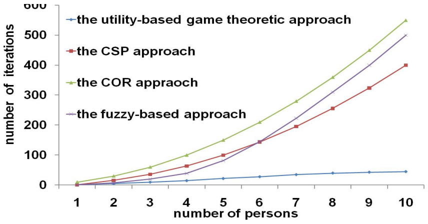

Figure X-6. Comparison for the number of iterations with a fixed number of cameras (10) and various numbers of the objects.

5. CONCLUSIONS AND FUTURE WORK

In this chapter, we analyzed existing and emerging techniques for the camera selection and handoff problem. Pros and cons of distributed and centralized systems are discussed. Four selected approaches are discussed in details. Both theoretical and experimental comparisons are provided. It is shown that the utility-based game theoretic approach is more flexible and has low computational cost. However, it is centralized unlike the CSP approach and the fuzzy-based approach. The COR approach is not applicable when the scenario is complicated.

There is the trend to have a hierarchical structure which hybrids the distributed and centralized control. There is a lack research on camera selection and handoff in a large scale network of active cameras. Current research is short on experimental results with real data processed in real time. Embedded systems are attracting increasing attention. However, the limitation of resources requires for more efficient software algorithms that can run on embedded systems reliably.

6. ACKNOWLEDGEMENT

This work was partially supported by NSF grants 0551741 and 0622176, 0905671 and ONR grants (DoD Instrumentation and the Aware Building). The contents of the information do not reflect the position or policy of the US Government.

7. REFERENCES

[1] Y. Li and B. Bhanu, " Utility-based Camera Assignment in a Video Network: A Game Theoretic Framework", IEEE Sensors Journal, Special Issue on Cognative Networks, 2010.

[2] H. Kim, C. Nam, K. Ha, O. Ayurzana and J. Kwon, “An algorithm of real time image tracking system using a camera with pan/tilt motors on an embedded system”, ICMIT, Proc. of SPIE Vol. 6041 60, 2005.

[3] M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof and B. Strobl, “Autonomous multicamera tracking on embedded smart cameras”, EURASIP Journal on Embedded Systems, 2007.

[4] B. Rinner, M. Jovanovic and M. Quaritsch, “Embedded middleware on distributed smart cameras”, ICASSP 2007.

[5] S. Flech and W. Straβer, “Adaptive probabilistic tracking embedded in a smart camera”, CVPR 2005.

[6] S. Flech, F. Busch, P. Biber and W. Straβer, “3D surveillance-A distributed network of smart cameras for real-time tracking and its visualization in 3D”, CVPR Workshop 2006.

[7] J. Park, P.C. Bhat and A.C. Kak, “A look-up table based appraoch for solving the camera selection problem in large camera networks”, Workshop on DSC 2006.

[8] K. Morioka, S. Kovacs, J. Lee, P. Korondi and H. Hashimoto, “Fuzzy-based camera selection for object tracking in a multi-camera system”, IEEE Conference on Human System Interactions, 2008.

[9] C. Micheloni, G.L. Foresti and L. Snidaro, “A network of co-operative cameras for visual surveillance”, Vision Image Signal Process., April 2005.

[10] F. Bashir and F. Porikli, “Multi-camera control through constraint satisfaction for persistent surveillance”, AVSS, 2008.

[11] S. Lim, L.S. Davis and A. Elgannal, “A scalable image-based multi-camera visual surveillance system”, AVSS 2003.

[12] V. Kattnaker and R. Zabih, “Bayesian multi-camera surveillance”, CVPR, 1999.

[13] Y. Lu and S. Payandeh, “Cooperative hybrid multi-camera tracking for people surveillance”, Can. J. Elect. Comput. Eng., Vol. 33, No. 3/4, 2008.

[14] I. Everts, N. Sebe and G. Jones, “Cooperative object tracking with multiple PTZ cameras”, ICIAP, 2007.

[15] S. Kang, J. Paik, A. Koschan, B. Abidi and M.A. Abidi, “Real-time video tracking using PTZ cameras”, Proc. of SPIE, Vol. 5132.

[16] Y. Li and B. Bhanu, “Utility-based dynamic camera assignment and handoff in a video network”, ICDSC 2008.

[17] O. Javed, S. Khan, Z. Rasheed and M. Shah, "Camera handoff: tracking in multiple uncalibrated stationary cameras," IEEE workshop on Human Motion, 2000.

[18] 2010M. Valera and S.A. Velastin, “Intelligent distributed surveillance systems: a review”, Vision Image Signal Process., Vol. 152, No. 2,2005.

[19] B. Moller, T. Plotz and G.A. Fink, “Calibration-free camera hand-over for fast and reliable person tracking in multi-Camera setup”, ICPR 2008.

[20] Y. Jo and J. Han, “A new approach to camera hand-off without camera calibration for the general scene with non-planar ground”, VSSN’06.

[21] C. Chen, Y. Yao, D. Page, B. Abidi, A. Koschan and M. Abidi, “Camera handoff with adaptive resource management for multi-Camera multi-target surveillance”, AVSS 2008.

[22] J. Nayak, L. Gonzalez-Argueta, B. Song, A. Roy-Chowdhury and E. Tuncel, “Multi-target tracking through opportunistic camera control in a resource constrained multimodal sensor Network”, ICDSC 2008.

[23] A. Gupta, A. Mittal and L.S. Davis, “COST: An approach for camera selection and multi-object inference ordering in dynamic scenes”, ICCV 2007.

[24] B. Song and A.K. Roy-Chowdhury, “Robust tracking in a camera network: a multi-objective optimization framework”, IEEE Journal of Selected Topics in Signal Processing, Vol. 2, No. 4, Aug. 2008.

[25] B. Song, C. Soto, A.K. Roy-Chowdhury and J.A. Farrell, “Decentralized camera network control using game theory”, ICDSC Workshp 2008.