Machine Vision and Applications
DOI 10.1007/s00138-007-0082-2

SPECIAL ISSUE PAPER

Coopetitive multi-camera surveillance using model predictive control

Vivek K. Singh · Pradeep K. Atrey · Mohan S. Kankanhalli

Received: 15 April 2006 / Accepted: 10 January 2007

Abstract We present a generic framework for enhanced active multi-sensing. We propose a *coopetitive* interaction approach, which combines the salient features of cooperation and competition with an aim to optimize the cooperation among sensors to achieve best results at the system level rather than redundantly implementing cooperation at each stage. We also employ model predictive control based forward state estimation method for counter-acting various delays faced in multi-sensor environments. The results obtained for two different visual surveillance adaptations with different number of cameras and different surveillance goals provide clear evidence for the improvements created by adoption of the proposed enhancements.

Keywords Visual surveillance · Coopetition · Model predictive control

1 Introduction

The benefits of active cooperative sensing as compared to non-cooperative sensing have been well established in literature [5,4]. By active cooperative sensing, we mean that the sensors in use not only react based on the sensed data, but they also help each other by exchanging information among them in order to better perform the designated task. However, such multi-sensor systems often face the following two key problems:

1. Lack of sophisticated interaction mechanisms which can optimize the cooperation among sensors to achieve best results at the system level rather than redundantly implementing cooperation at each stage.
2. The delay in exchange of information among sensors and the delay in sensors reacting to the received instructions.

We justify the importance of the above two issues as follows. Let us consider a dual camera surveillance scenario where the system goal is to obtain high resolution images of an intruder entering an enclosed area. In this scenario, the camera which is in a better position to capture the images of intruder should be allowed to track him/her even if it means competing against the peer camera in certain instances e.g. based on the size of facial image obtained. Hence in this case even though the cameras are 'competing' in a local context (i.e. in capturing the images of intruder), they are still 'cooperating' toward a common goal in a global context by working together to obtain the best high resolution images of the intruder. Therefore, there is a need for employing a more sophisticated interaction strategy which involves 'competition' as well as 'cooperation' among the sensors instead of having just redundant cooperation.

Also, there is always some delay encountered in information exchange among sensors and there is inherent latency in sensors reacting to any obtained information. This delay significantly reduces the system's speed and in turn the performance. For instance, in camera-based systems, all tracking based applications face a single frame delay between observing certain traits and reacting to it [21,22]. Also, all multi-camera systems observe a significant delay if PTZ (Pan, Tilt and Zoom) parameters are passed between cameras

V. K. Singh (✉) · P. K. Atrey · M. S. Kankanhalli
School of Computing, National University of Singapore, Singapore, Singapore
e-mail: vivekkum@comp.nus.edu.sg

P. K. Atrey
e-mail: pradeepk@comp.nus.edu.sg

M. S. Kankanhalli
e-mail: mohan@comp.nus.edu.sg

to undertake any specific task. Hence, there is a need for a better control mechanism to counter the delay problem.

The above arguments clearly establish the need for an effective framework which can create synergistic interactions between sensors to achieve system level cooperation and at the same time offset delays which occur frequently in practical implementations of active sensing systems. In this paper, we propose the use of 'model predictive control (MPC)' feedback mechanism coupled with a 'coopetitive' interaction strategy to address these issues, and thereby enhance the performance of various multi-sensor systems.

MPC is an effective control mechanism that allows sensors to react not only based on past observations but also estimated future events. In effect, this control method counterbalances the delays by making explicit use of the process model to predict the system control inputs and the outputs. It has been widely used in robotics and chemical engineering fields [1,10] and has been clearly shown to give better performance than non-estimation based control mechanisms like PID (proportional integral differential)[10,19]. To correlate MPC to a real-life example of chess; we play our next move not only based on past opponent moves but also on his/her anticipated future moves.

To enhance the performance of multi-sensor systems, we also need to adopt the concept of *coopetition* among the sensors. We define *coopetition* as a process in which the sensors 'compete' as well as 'cooperate' with each other to perform the designated task in the best possible manner. In other words, we propose the use of cooperation based on an explicit notion of worthiness or merit as decided by competition. To correlate it to a real-life example of a soccer match, a mid-fielder would pass the ball to a striker of his team only if the striker has a better ability to score a goal, else he would try the shot himself. Hence, even though the two players would be 'competing' to score a goal at a local instance, they are still 'cooperating' globally to optimize the overall task of scoring as many goals for their team as possible.

We generalize the definition of cooperation to include the concept of transfer of *roles and sub-goals* rather than just parameters as often described in sensor literature. Also, such passing of roles and parameters should be done only when there is an explicit need to do so ('interact on demand') rather than at each instant as there are significant overheads required for such transfers. Thus we believe that sensors should not only pass useful information (as in [25,13] etc.) to each other but also pass over their entire strategy and role to the other sensor if it helps in achieving better overall performance. Hence, we introduce the ideas of *dynamic role exchange* and *interact on demand* between sensors for achieving global optimization. Another real life example to clearly describe our over-all *coopetitive* interaction strategy can be seen at the popular card game of bridge in which partners try to outbid each other in an attempt to obtain best possible results for their team. We also notice that partners do not just pass each other parameters but rather give up their role (e.g. as a bidder) if they realize that doing so will lead to over-all best results for the team.

We have earlier described preliminary ideas of 'coopetition' interaction strategy and MPC for a dual camera surveillance setup in [22]. This paper provides a detailed discussion on how a generic framework with m sensors, n sub-goals and any specific system goal can benefit from the use of MPC and coopetition. We shall, also undertake a comprehensive discussion on how such a proposed framework can be scaled to different surveillance systems with differing goals and number of sensors.

As stated, our proposed framework is generic and can in-principle be applied to any number of sensors/cameras and to any system goal. Hence, to demonstrate its versatility, we describe two sets of experiments with differing number of sensors as well as the system goals. Both the implementations have been chosen from the surveillance domain, which is indeed an important area in multi-sensor research. The first set of experiments demonstrates the application of a dual camera system to capture high resolution images of an intruder entering an enclosed single door rectangular premises e.g. an ATM lobby or a museum sub-section.[1] The second set of experiments demonstrates the use of a triple camera system for the detecting any abandoned objects in any well defined surveyed premises. Specifically, we are interested in capturing facial images of the person abandoning the object, the object itself and the trajectory of the person.

The two sets of experiments allow us to obtain quantifiable measures of the performance of our proposed approach. The results obtained are compared with the base case scenarios like those of static sensors or *non-coopetitive* interactions and also contrasted with those of other approaches described in literature.

To summarize, our main contributions in this paper are:

1. Adoption of *coopetitive* interaction approach between sensors in order to achieve the overall system-level optimization. We have established through experimental results in a surveillance scenarios that this interaction strategy helps us to assign the best-suited visual sensors for the task at hand and achieve best possible overall results.
2. Introduction of MPC (model predictive control) as a novel means of handling feedback in active sensors. The ability to react to the possible future scenarios gives us an effective tool to offset transmission and reaction delays which often affect the performance of multiple visual sensor systems.

[1] Results for some of the dual camera setup experiments were described earlier in [22].

The organization of the remainder of the paper is as follows. In Sect. 2, we describe the related work done by the research community. Section 3 presents the detailed description of our framework and also provides a theoretical analysis of the various tools and techniques used. Section 4 describes the practical implementation details for the two scenarios and in Sect. 5, we describe the results and perform a detailed analysis. Section 6 summarizes the work done and the contributions with a discussion on the future work.

2 Related work

As stated earlier, the main contributions of this paper are regarding the mode of interaction (i.e. *coopetition*) among sensors and introduction of MPC as a novel feedback mechanism. Hence we discuss the related work in these two areas and also take a closer look at similar dual and multiple camera surveillance frameworks.

From a multi-agent robotics perspective, Stentz [7] describes cooperation and competition between robots in order to fulfill assigned robotic tasks e.g. clearing toxic waste etc. Bowyer [2] also gives a good description of different types of cooperation between sensors from generic point of view. The ideas described in these papers are very interesting, however they are from a robotics/multi-agent perspective with no correlation to visual surveillance or vision sensor interactions. In our current work, we aim to describe the effective use of a combination of cooperation and competition for obtaining best performance in vision-based systems.

Laurent [12] describes the use of an omni-view camera combined with a laser sensor for object localization. This work however does not deal with the interaction strategies required for cooperation. Molina et al. [8,3,15] describe some interesting strategies for coordination between cameras for surveillance. They advocate prioritization between sensing tasks and also touch on the concepts of conflict management and hand-off. However, they do not propose the explicit use of competition between sensors as we propose in our coopetitive framework. Nor do they talk about the use of competition and cooperation *only* when required as we propose. They also do not deal with forward state-estimation which is an integral part of our framework.

On the other hand, Barreto et al. [1] have done some work on feedback control mechanisms for vision based systems. They have highlighted the use of MPC mechanism to control the motion of a camera placed in robot head. They also use a Kalman filter for predicting the future positions of the tracked object so as to handle system delays. Papanikolopoulos et al. [18] also proposes the use of MPC and Kalman filter to track objects for a camera placed in the robot head. Saedan et al. [20], on the other hand describes the use of PID control for visual tracking by robot head camera. However, these works deal with single camera robotic systems and do not cover the interaction between multiple cameras and the complexities arising such as delay and the competition/cooperation issues. We intend to create an integrated system which would have an efficient feedback mechanism combined with the ability to handle interactions across multiple cameras.

VSAM (visual surveillance and monitoring) project [4,5] describes the concept of cooperation among multiple active cameras for tracking objects. It also provides a very good overview of the visual surveillance area. However it does not deal with the specific issues such as *coopetition* and delay counter-action which we are handling.

Recently, some interesting dual-camera frameworks have been proposed. Collins et al.[25] describes a master-slave approach to detect human beings at a distance. In this system, the master camera takes wide panoramic images and the slave camera zooms into the person to obtain his images. Regazzoni et al.[14] also describe a similar dual camera setup for obtaining focused images of objects at a distance. Liu et al. [13] also adopt a similar master-slave approach. Their system has a wide-angle panoramic camera with a PTZ (Pan Tilt Zoom) camera on top. Anastasio et al. [23] describe the use of a wide-angle camera combined with an active camera to obtain human images. The work by Greiffenhagen et al. [9] uses an omni-view camera attached to a PTZ active camera to obtain images of the monitored object. However, in all these works, there is no movement of the master camera and both cameras are placed facing the same direction hence reducing the possibilities of obtaining good quality images of the monitored object in all cases. For example, a change of face direction by the human being will cause these systems to lose out on his facial information.

As mentioned earlier, we have described the preliminary ideas on coopetitive approach and the results with two interacting cameras in [22]. In this paper we have generalized the approach to a framework with m sensors, n sub-goals and any specific system goal. We have also bolstered the experimentation for dual camera scenarios and studied the adaptation to a triple camera scenario in detail. Thus the aim of this paper is to provide a generalized framework which employs coopetitive interaction and MPC feedback and describe detailed experimentation results for the same.

A summary of related work has been shown in Table 1. It clearly highlights the attributes of visual surveillance which have already been adopted by the research community and also those which have been proposed for the first time in this paper.

As can be seen in Table 1, a significant amount of work has already been done using multiple active cameras. Interaction between cameras is also commonly described, but the *coopetitive* approach of interaction between cameras for global optimization has been introduced for the first time in our proposed framework. MPC has been described from

Table 1 Summary of different attributes across various related work

Work	Active cameras	Multiple cameras	Interaction strategy	Model predictive control	Dynamic role swapping	Interact on demand
Stentz [7]	–	–	Cooperation and competition	–	–	–
Molina [16,8,15]	Yes	Yes	Coordination	–	Hand–off	–
Barretto [1]	Robot head movement	–	–	Yes	–	–
Papanikolopoulos [18]	Robot head movement	–	–	Yes	–	–
Saedan [20]	Robot head movement	–	–	–(PID)	–	–
Collins[4,5]	Yes	Yes	Cooperation	–	–	–
Liu [13],Collins [25] Regazzoni [14], Anastasio [23], Greiffenhagen [9]	Yes	Dual camera	Cooperation	–	–	–
Proposed framework	Yes	Yes	*Coopetition*	Yes	Yes	Yes

robotics perspective in a couple of works but its prowess for visual surveillance has been highlighted for the first time in on our works. While something similar has been hinted in one work, the concept of 'dynamic role exchanging' has been explicitly adopted for the first time in our proposed framework. Lastly, the concept of 'interact on demand' has also been highlighted for the first time in this paper.

3 Proposed framework

We propose a generic framework which can be employed for accomplishing any well defined task with m sensors and n sub-goals, followed by its adaptation for the two and three camera surveillance scenarios. We also discuss on how the proposed novel methods of coopetitive interaction and MPC work and how they have been implemented in our systems.

3.1 Generalized framework

The algorithm for our proposed generic framework has been shown in Fig. 1. The process of coopetitive interaction among sensors is initiated based on a trigger event. This trigger event can be 'a person entering a room' or 'an object being abandoned' or even 'a traffic blockage occurring on the highway'. Upon triggering, the framework starts assigning each of the n tasks or sub-goals to each of the m available sensors. The most important and/or most restrictive task is assigned its sensor first, followed by the second most important task and so on. The assignment of each of these sensors needs to be based on an explicit notion of suitability or worthiness amongst its peers. This phase constitutes the 'competition' phase of over-all coopetitive strategy.

Upon allocation of the appropriate tasks to the sensors, any sensor which has information that may help other sensors passes this information to the appropriate sensor(s). This information could be the last known position of an intruder in the room, or the position of an abandoned object, and so on. This exchange of information would be especially useful in subsequent iterations when the roles might be exchanged among sensors. This phase constitutes the 'cooperation' component of our coopetitive framework.

Once the roles are allocated and the useful information is obtained, the sensors undertake their respective tasks. The system keeps track of the performance each sensor, as the sensors may not always be able to undertake the allocated task well. For example, when the intruder might change his facial direction, another sensor may be better equipped to focus on his face. If the sensors are not able to achieve their intended goal, we consider re-allocation of tasks. However, this re-allocation is done only when it is appropriate to do so, e.g. the re-allocation of tasks when the face is not found for just one frame may not be optimal as it may be due to incorrect face detection. Similarly, it may not make sense to re-allocate tasks if some of the sensors have already started working on their assigned task and are no longer in a position to undertake any other roles. On the other hand, if the sensors are able to undertake their allocated tasks correctly, they continue working on their allocated tasks until all their tasks or the sub-goals are completed.

In this framework, we make a basic assumption that $m \geq n$ i.e. we have more sensors (m) than the tasks (n), at any given instance. This is important as it is usually not practical to assign more than one task to a sensor at the same time. The roles can however be exchanged among sensors as and when required. Hence, even if we need all n tasks to be undertaken from each sensor location, we do not need to employ $n \times m$

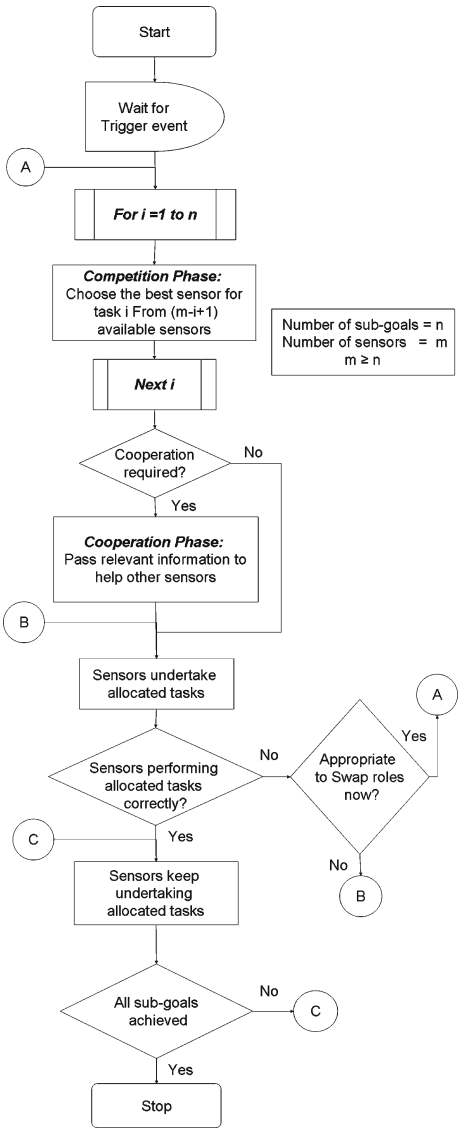

Fig. 1 Generic algorithm of the proposed framework

different sensors; but rather just allocate different roles to the same sensor at different points of time. Another assumption made is that the system designer can indeed prioritize the constituent sub-goals for any particular adaptation of the generic framework.

Having covered the generic algorithmic aspects of the framework, we now take a closer look at how this framework can be adapted to a specific dual camera and a triple camera system, in the two subsequent subsections.

3.2 Dual camera system

For the dual camera adaptation, we consider a scenario of monitoring the intruders entering into a single door enclosed rectangular environment e.g. an ATM lobby or a museum subsection. The system goal is to obtain high quality frontal facial images of the intruder. These high quality frontal images can be useful for further automated processing e.g. face recognition etc.

One of the two cameras in the system undertakes the task of observing the entire room in order to detect new intruders, and the second camera focuses (i.e. tracks and zooms) onto their faces.

The algorithmic approach for the proposed framework has been illustrated in Fig. 2. The triggering event for the dual camera system is the entry of a person into the room. In the competition phase, both the cameras try to focus on to the face, but only the more suitable camera is allowed to undertake this task. This role allocation is done based on an explicit notion of merit between cameras. The measure of merit adopted is un-zoomed size of facial images (if any) obtained by cameras.

Following this, a decision is made as to whether any cooperation or passing of roles and parameters is required. This role-exchanging is not required in first cycle but shall be

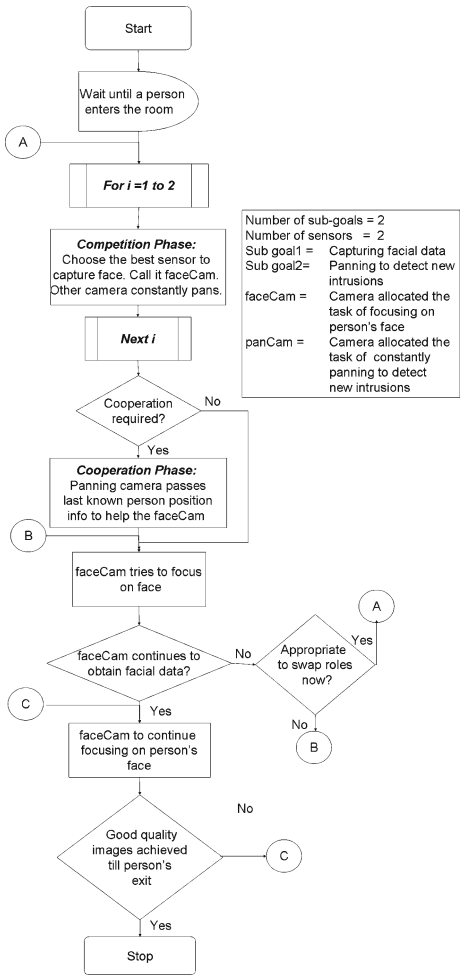

Fig. 2 Algorithm for the dual camera surveillance adaptation

required in later cycles when a camera loses facial image as the person has turned in the other direction. This process again requires an explicit measure of suitability to transfer roles at that particular instant. For our case if the faceCam (i.e. the camera focusing on the face) cannot find any faces even after zooming out to increase field-of-view, then it assumes that person has changed direction. The faceCam continues to focus on person's face until he exits the surveyed premises.

A more detailed explanation for this dual camera system can be found in [22].

3.3 Triple camera system

For the triple camera adaptation of the framework, we consider a surveillance scenario where we monitor a walk-way for any abandoned objects. If the system detects any abandoned objects, it obtains the images of the person abandoning the object, the object itself and the trajectory. Specifically, we define our three system sub-goals as:

1. Obtaining three 'good quality' images of the intruder: We define 'good quality' images as the frontal facial images of the intruder with at least 100 by 100 pixel resolution. Images at this resolution can be used automated face recognition procedures with an accuracy of around 90% [24] and are thus useful for terrorist identification etc.
2. Obtaining three 'good quality' images of the object: The good quality images for objects are the focused zoomed images with at least 100 by 100 pixel resolution. This allows the security personnel who might be receiving these images to have a much better idea about the possible nature of this object before taking their next action.
3. Obtaining the trajectory of the person: This gives us a better over-all picture of the scenario as well as provide further insight into the motives of the person who abandoned the object.

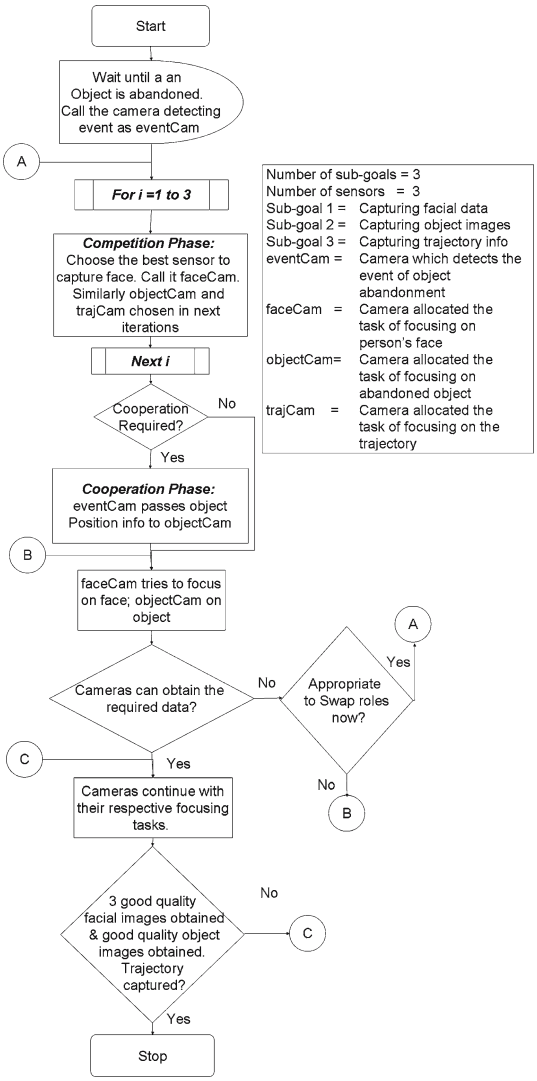

Fig. 3 Algorithm for the triple camera surveillance adaptation

The overview of the adapted algorithm has been shown in Fig. 3. The triggering event for this surveillance scenario is the abandonment event. We detect the event based on blob detection using the splitting of one moving blob into one moving and one static blob as the symptom of abandonment event. All three cameras keep looking out for this event and the camera which successfully detects the event keeps a log of the object position for future reference.

The coopetitive framework starts with the assignment of roles to all three sensors based on the three system sub-goals. The camera which is best suited for face capturing is allocated this task and is labeled as 'faceCam'. We choose the faceCam before choosing other cameras because capturing the face is the most restrictive task amongst the three sub-goals. Only the camera towards which the person is facing shall be able to this task effectively. The tasks of focusing on object or watching for the trajectory on the hand be handled well by more than one cameras. Hence, we iterate and choose the 'objectCam' to focus on the object and the 'trajCam' to observe the trajectory in the second and third iterations respectively. The measure of merit for the allocation of trajCam is the distance between the person blob and the object blob as we do not want the person to block the object images as we zoom in to capture the object images. This merit-based task allocation part constitutes the competition phase of the over-all coopetitive interaction strategy.

Next phase is the cooperation in which the previously recorded object position information is passed to the objectCam. This information is useful to objectCam as the objectCam may not always have the object in its field of view. Furthermore, even if it can see the object or/and the person, it

may not be able to independently differentiate between them and decide which one is an object.

In this adaptation, since we want to obtain just three images of the person, there is a reasonably high probability that the allocated camera chosen based on the merit of un-zoomed face size in its images can undertake the task efficiently. As such we do not employ swapping of roles in this adaptation since both the faceCam and objectCam would no longer be at their initial position, and hence, the parameter transfer may not be appropriate after they have started performing their respective tasks. Also, the trajCam should not be moved as it is the only camera which captures the over-all scenario as well as the trajectory. Transferring this role halfway to another camera is not appropriate. The cameras keep undertaking their respective tasks until the three sub-goals have been achieved.

The three cameras have been arranged in triangular placement. An overview of the arrangement of the cameras has been shown in Fig. 4.

Some of the assumptions which we have employed in this implementation are as follows:

1. Similar to the dual camera experimentation, we consider person's face as an important region of interest. Hence, we assume that the person is not wearing any mask etc to hide his face.
2. We assume that the person's frontal face is visible in at least one of the camera.
3. We also assume the system sub-goals can be linearly arranged in an order. This arrangement is used to assign the cameras to the sub-goals. Of course, the first sub-goal has the luxury of choosing the most appropriate sensor from the m available sensors while the second sub-goal can choose the best available from $m - 1$ sensors and so on. For our implementation purpose, we have chosen the most restrictive and important task of face detection as the first priority sub-goal.

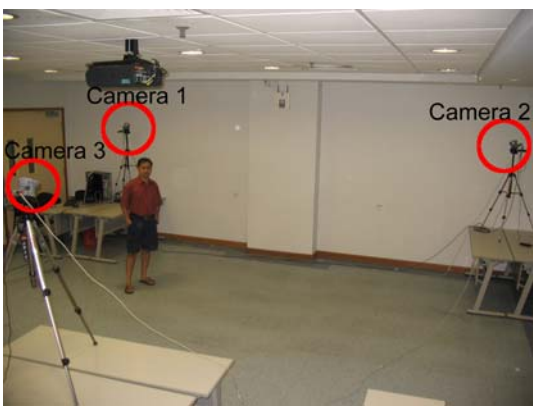

Fig. 4 Overview of the camera arrangement for three camera setup

3.4 Use of *coopetitive* interaction approach

We call our framework's interaction approach as *coopetitive* in the sense that the cameras both compete and cooperate for efficient visual surveillance. However, it is important to note that the competition we are dealing with is intra-team i.e. the competing entities still share a common overall goal (e.g. members of same team in a bridge card game trying to outbid each other) as opposed to inter-team (e.g. members of opposite team in a bridge card game) in which case the entities may have opposing system goals. This type of coopetitive interaction takes place in both the dual-camera and the triple camera surveillance scenarios.

In the dual-camera setup, initially the cameras compete against each other to undertake the role of focusing on the monitored object (here after called M_{Obj}). In a single intruder case, this competition is clearly won by the camera towards which the intruder is facing. However, this competition becomes more interesting in the multiple intruder scenario where the winner of competition must be decided based on a measure of merit. In fact, our aim is to just provide a generic notion of 'competition' which can support measures of merit of all varieties. It can be a simple decision based on face size in images or a complex function based on highest resolution facial image of person obtained as yet, room parameters, lighting conditions and so on. A measure of discrimination which gives higher priority to new intruder's faces is also plausible. However, for the purpose of the experiments described in this paper, we use the size of un-zoomed face as a measure of merit to discriminate between sensors.

Let us say, without the loss of generality, the camera C_1 wins the initial competition and starts focusing on the OOI (object of interest). Then the camera C_2 starts panning the entire area searching for new OOIs. However, if at a later point of time C_1 can not obtain OOI images anymore e.g. due to change of facial direction by intruder, it would cooperate by passing over its role as well as the information regarding possible location of the OOI to C_2. Hence our cameras both compete and cooperate at different moments of time to allow the best suited sensor to take over the task of obtaining OOI images.

In the triple camera setup, the cameras initially compete with each other to obtain the most important role as faceCam. However, only the most deserving camera is allocated this task. Once again, while our generic framework does not necessitate any specific measure of merit, for implementation purposes we have chosen the size of un-zoomed facial images obtained by different cameras as a measure of merit. Similarly, for the second sub-goal we use the Euclidean distance between person and the object blobs as the measure of merit.

Also, at the moment the triggering event of abandonment is detected, the camera doing so keeps a record of the position

information of the object blob. This information is passed over at a later instant to the allocated objectCam to help it in focusing on the object. Hence, we realize that while the different sensors initially are 'competing' to obtain the various tasks, they 'cooperate' later on to help improve the over-all system performance. Hence we allow the best suited cameras to undertake the tasks and still allow them to leverage on to the available help from other sensors.

We also notice that the role-allocation is dynamic in both the setups. The sensors employed are equally capable and the roles can be exchanged or swapped as the situation may need. For example, in the triple camera setup, initial task of detecting the abandonment event can be accomplished by any of the sensors. Even later on, the faceCam and the objectCam tasks can be allocated to any of the cameras. In fact, the camera allocation can be totally different for the same person entering the premises twice depending on his trajectory chosen.

In the dual camera setup, we notice this trait even more prominently. The roles between the equally capable PTZ cameras can be swapped at any time, if doing so improves the system's over-all results. This provides better performance results for face imaging tasks as compared to other dual camera systems like [23,25,13], which employ two cameras to detect faces in only one direction or would need four cameras to obtain person data in both the directions.

3.5 Use of MPC

We have adopted the MPC feedback mechanism for both our dual and triple camera systems. In dual camera system, the use of MPC allows us to better estimate the position of the walking person by the time the new faceCam is able to focus onto it. It also helps us in tracking the person's face as we use MPC to predict the frame position of the face rather than being lagged by one frame in each iteration. In triple camera setup also, the use of MPC has allowed us to better track the person's face in order to focus on to it to obtain high quality images. In this subsection, we describe the theoretical background and reasoning behind MPC, and also how it has been implemented in our systems.

The working of the MPC has been illustrated in Fig. 5 adapted from [17]. At time t, based on the past and future (estimated) values, the system tries to decide optimal values of manipulated inputs $u(t+k)$. However, only one input $u(t)$ is actually fed into the system. The same process is repeated at time $(t+1)$ i.e. based on the input and output values till time $(t+1)$, future values of manipulated input and predicted output are decided. Such a process is repeated at the end of each time interval in the duration of interest i.e. till time $(t+p)$.

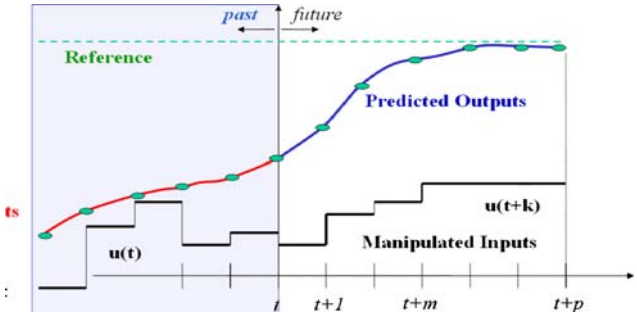

Fig. 5 Working of MPC

3.5.1 Our MPC framework

The MPC framework adopted by us for surveillance has been divided into four parts as shown in Fig. 6. The *input* to the system is the movement to be made by the camera in order to try to bring the M_{Obj} centroid to the center of the image plane. The reference point for the signal is the center of the image plane and the *output* of the system is the actual position of the M_{Obj} centroid obtained on the image plane.

The aim of our framework is to obtain the images with the M_{Obj} centroid placed at the center of the image plane. In order to achieve this, the framework works as follows. The system dynamics (Part A) is responsible for converting the system input i.e. control signal in terms of image plane $[x, y]$ coordinates into pan and tilt movement parameters for the camera. Based on this camera movement, we measure the M_{Obj} centroid position as the output of the system. However, in MPC, very often we try to predict the input and output data for future instances in order to achieve global optimization. Under such circumstances, we use a state estimation mechanism (Part B) to estimate the future M_{Obj} centroid positions. We are using a Kalman filter approach to estimate the future M_{Obj} positions.

Part C refers to the reference point which is the center of the image plane. After obtaining the actual/estimated M_{Obj} positions for the duration of interest and comparing it with

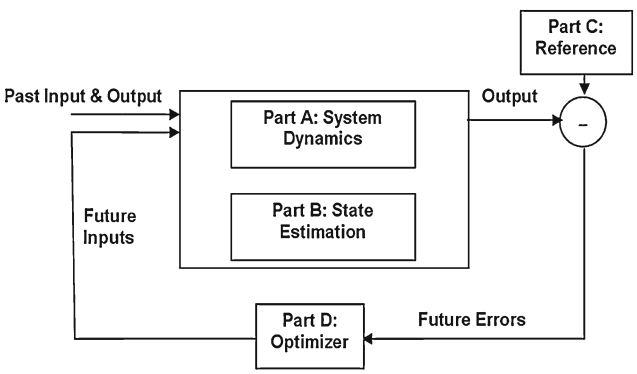

Fig. 6 The proposed MPC framework for visual surveillance

the Reference value we are able to obtain the error values in terms of $[x, y]$ coordinates. This value is passed to the optimizer (Part D), which decides on the optimal control signal to be sent at the current instant so as to achieve an overall minimum value for the penalty function. The penalty function is a weighted average of the estimated error and the control effort required. The above mentioned process is repeated at the end of each frame in an effort to bring the M_{Obj} image centroid to the center of the image plane.

Now let us look at each of the four parts of our MPC framework mentioned above in more detail.

A: System dynamics

The system dynamics part includes the conversion of the input control signal in term of $[x, y]$ parameters into pan and tilt angle values which are implemented on the camera. The relation between $[x, y]$ coordinate deviations and the corresponding angles can be found using camera calibration. In our particular framework using Canon VC-C4 cameras with 384 by 288 pixel resolution frames, we found the pixel to angle ratio to be 16 pixels/degree in both horizontal and vertical directions.

B: State estimation

The process of state estimation is required when we need to estimate the M_{Obj} positions for future instants of time. We have adopted the widely used [5,1,11] Kalman filter approach for the purpose of state estimation. We have assumed a constant velocity model to undertake state estimation and modeled system noise as the difference between measured and predicted values at current time (t).

Our overall equation to calculate the M_{Obj} position for next time instant i.e. (t+1) is

$$y_o = \begin{bmatrix} p_x \\ p_y \end{bmatrix}_{t+1} = \begin{bmatrix} p_x \\ p_y \end{bmatrix}_t + \begin{bmatrix} T \\ T \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}_t + G \begin{bmatrix} px_t - px_{t|t-1} \\ py_t - py_{t|t-1} \end{bmatrix} \tag{1}$$

y_o is the optimal estimate, p_x is the position on x axis, p_y is the position on y axis, T is the time duration between frames, v_x is the x-axis velocity, v_y is the y-axis velocity, G is the Kalman gain, px_t is the x-axis position of M_{Obj} as measured at time t, $px_{t|t-1}$ is the x-axis position of M_{Obj} at time t as predicted at time $t-1$, py_t is the y-axis position of M_{Obj} as measured at time t, $py_{t|t-1}$ is the y-axis position of M_{Obj} at time instant t as predicted at time $t-1$.

C: Reference

In our framework, we want the M_{Obj} centroid to be imaged at the center of the image plane. This allows high quality facial images to be obtained in the center of the image plane together with contextual information from the non-center portions. Our reference point always remains at the center of image plane [0, 0] assuming that egomotion i.e. the motion of the camera itself, has been compensated for. The compensation for egomotion is handled by the optimizer section when it makes calculations on appropriate input parameters to be fed into the system.

D: Optimizer

The basic aim of the optimizer is to find out the optimal current input (u), which decreases the M_{Obj} tracking error as well as the control effort required. Hence, we want to minimize a penalty term that represents the weighted sum of future errors and control actions. This can be represented in a mathematical form as follows:

$$\min_{\Delta u} \; Q \sum_{k=N_1}^{N_2} (y(k) - \text{ref}(k))^2 + R \sum_{k=N_1}^{N_2} (u(k+1) - u(k))^2 \tag{2}$$

where, Q, R are the weight factors, k is the instant of time being considered, N_1, N_2 are the start and end point of the duration of interest, $y(k)$ is the output i.e. M_{Obj} centroid on the image plane, $u(k)$ is the control input i.e. movement of camera in terms of image plane coordinates, $\text{ref}(k)$ is the reference signal i.e. image plane center [0,0].

The factors Q and R decide the relative importance given to the reduction of future error and the control action. After a few rounds of experiments and tuning, the values of 0.8 and 0.2 were found to be most appropriate for parameters Q and R respectively. This is due to the fact that in our framework reducing tracking error is much more important than reducing the camera movement.

The $y(k)$ as mentioned in Eq. (2) is the position of M_{Obj} centroid on the image plane and can be obtained for future frames by using Kalman filter as described in Eq. (1). The $\text{ref}(k)$ represents the center of the image plane [0, 0], but it needs to be compensated for the movement of the camera itself. Hence for future instants of time:

$$\text{ref}(k) = \Delta u(k) \tag{3}$$

This process of parameter translation across frames has been demonstrated in Figs. 7, 8. While Fig. 7 demonstrates the normal error as observed for two consecutive frames in the case of no camera movement between frames, Fig. 8 demonstrates how these parameters are translated in the next frame in case there is camera movement between frames. Basically

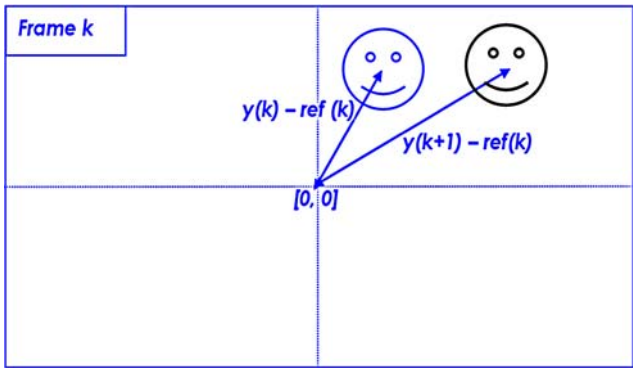

Fig. 7 Movement of M_{Obj} centroid on the image plane between two consecutive frames

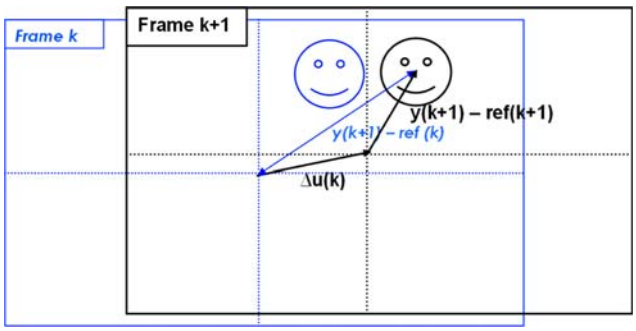

Fig. 8 Impact of control input on the reference frame

the reference itself moves by $\Delta u(k)$, and hence the effective error should be reduced by $\Delta u(k)$ to compensate for the camera movement.

The second term of Eq. (2) clearly represents the movement of the camera between frames and can be written as $\Delta u(k)$ too. Hence, Eq. (2) can also be written as:

$$\min_{\Delta u} \; Q \sum_{k=N_1}^{N_2} (y(k) - \Delta u(k))^2 + R \sum_{k=N_1}^{N_2} (\Delta u(k))^2 \tag{4}$$

To facilitate the solution of this equation we bring it to a standard quadratic form, which is

$$\min_{x} \; \mathbf{x}' H \mathbf{x} - g' \mathbf{x} \tag{5}$$

For which the solution is described in [17] as $\mathbf{x} = 0.5 g H^{-1}$. We translate our MPC problem of Eq. (4) to the standard quadratic form given in Eq. (5). The process of translation and simplification is similar to one described in [17]. The specific mathematical steps have been omitted due to space constraints. After simplification we obtain the final value of input parameters as:

$$\Delta U(k) = \frac{Q^2}{Q^2 + R^2} Y(k)_{n \times 2} \tag{6}$$

where, $\Delta U(k)$ is a matrix containing optimal values of next n incremental movements to be made by system at time k, $Y(k)$ is a matrix that represents the next n estimated M_{Obj} positions at time k, Q and R are the weight parameters as described earlier.

We use the above equation to obtain the values of $\Delta U(k)$ matrix after each iteration. The first value of this matrix is actually fed into the system and after this $\Delta U(k+1)$ is calculated at the next iteration. The first value of $\Delta U(k+1)$ matrix is fed into the system and such a process is repeated for the entire period of interest.

3.6 Choosing the appropriate time for role transfer

An important component of our framework is dynamic role swapping so as to continually employ the best suited sensor for the (sub) goal. Delayed role-swapping may degrade the system performance, however frequent changing of roles may also cause heavy overheads. Also, this appropriate sensor to sub-goal allocation problem is dependent on the adversary behavior which further compounds its complexity. The key idea is that the system has to continuously 'out-guess' the adversary to make the correct sensor to sub-goal allocation.

One way to look at this problem is to compare it with the cache-selection problem in memory/systems research. Only a limited number (say k) out of the n possible candidates can be chosen to be in the cache. Normally k is lesser than n and there are significant penalties for cache-misses. Similar problem occurs in surveillance where the system can monitor only k states out of the n possible states which the adversary can select. Also, just like cache-selection problems 'thrashing' remains a major issue here as a determined adversary may choose states just after they have they have been unselected (or unmonitored) by the system. However, unlike memory systems we cannot employ assumptions like spatial or temporal locality to make the better selection decisions. In fact we cannot make any assumptions regarding the adversary behavior other than the fact that he wants to beat the system.

This translates into an 'out-guessing' conflict and both sides want to out-guess each other. Conventionally this 'burden of out-guessing' has been borne by the system. Further, the most common strategy used is the reactive 'trailing' of intruder states. But there always exists some physical time-delay between the system realizing that it needs to change its state and the actual time at which it becomes functional in its new state. In practice, any determined adversary can exploit such a delay ($T_{\text{swapdelay}}$) and keep changing his states very frequently.

We propose a simple counter-strategy for the system. The system can choose to pass the 'burden of out-guessing' on to the adversary rather than taking it on itself. Instead of following any regular patterns or a reactionary approach towards state selection, the system can use a random approach itself. If the adversary does not want the system to win, he has equal (if not more) interest in trying to guess the state to be chosen

by the system in next time instant. Thus now the 'burden of out-guessing' lies squarely on the adversary rather than the system.

We also realize that despite its benefits, a truly random approach cannot be employed by any surveillance system as it needs certain performance bounds. Hence we impose certain parameter bounds to the proposed random approach so that the system can keep the adversary guessing but still provide a certain specified performance on average.

Such a random approach can be applied in any given n choose k scenario e.g. 2 cameras which can monitor any 2 out of the 4 directional (north, east, west, south) states which the adversary can take or any other similar scenario. However, we continue further discussion with the base case scenario of 2 choose 1 states as it occurs in our two camera adaptation. Clearly the adversary can choose two different states for his facial direction. But we can only allow camera in one direction to focus on the face (as the other camera must be reserved for detecting newer intrusions). Thus we want to study how we can better perform the sensor to sub-goal allocation at each time instant.

To understand the proposition further let us consider a scenario where the intruder has already turned his face direction a few times and wants to change his face direction once more so as to avoid face detection.

Based on our random approach we ensure that camera keeps focusing on the intruder for duration T_{persist} even if he has changed his facial direction. This proposition rests on the fact that the intruder also has limited number of states to choose from. Further, he is also aware that system may choose to follow him to his new state, thus there is a possibility that the adversary may want to return to his initial time within a certain amount of time.

Let us make the value T_{persist} to be random number R times larger than $T_{\text{swapdelay}}$. Hence,

$$R = \text{random}(0, v) \tag{7}$$

$$T_{\text{persist}} = R \times T_{\text{swapdelay}} \tag{8}$$

where parameter v gives a bound to maximum amount of time the system can continue to focus in the same direction even if the intruder has changed his direction. This value can not be too low as then it becomes a conventional reactive 'trailer' system, nor can it be too high as then the system runs the risk of not obtaining good quality images for a long time of a person who has genuinely changed his direction and continues walking in that direction. Based on some basic parameter tuning experimentation, in our framework, we have chosen v to be 6. Hence the system continues to maintain the same sensor to sub-goal for a random duration between 0 to 6 times of $T_{\text{swapdelay}}$ i.e. an average value of 3 times.

To illustrate the above-mentioned approach further, let us look at a simple example. We assume:

$T_{\text{swapdelay}} = 1\,\text{s(average)}$
$R = 3\text{(average)}$
$T_{\text{persist}} = 3 \times T_{\text{swapdelay}} = 3\,\text{s(average)}$

We know that the intruder can escape detection, only if he changes his facial direction during the parameter transfer time. If he changes back his direction before parameter transfer, the camera in the other direction is still active and shall capture his face. If we allow the parameter transfer to be completed before changing direction then he will get captured in his current position itself. Thus, assuming total random distribution of the time moments when the intruder changes his facial direction:
Prob(Escape) after 1 cycle =

$$\frac{T_{\text{swapdelay}}}{T_{\text{persist}} + T_{\text{swapdelay}}} = \frac{1}{(1+3)} = 25\%$$

Prob(Escape) after n cycles =

$$\left(\frac{T_{\text{swapdelay}}}{T_{\text{follow}} + T_{\text{swapdelay}}}\right)^n = \left(\frac{1}{4}\right)^n$$

For example, if the intruder stays in the surveyed area for 12 s (3 cycles) his probability of escaping is 1.56% i.e. probability of his image getting captured is 98.44%. Please note that theoretically the same probability of capture is possible with a fixed persistence time of 3 s. In practice however, a determined adversary can easily out-guess a system which is following a fixed pattern. Thus it makes sense to always keep the intruder guessing about the next time the system changes sensor roles. We shall further verify the applicability of the proposed random approach in the experiments Sect. 5.2.

4 System implementation

This section describes the practical implementation details for the two surveillance scenarios adopted.

In both the adaptations, we have made use of Canon VC-C4 PTZ (Pan Tilt Zoom) capable cameras. Conceptually we treat each camera as a separate entity which can compute its own suitability metric and in turn be granted a task. The practical implementation has been done with the use of a VC++ software running on a central PC which has separate objects of the camera-agent class for each camera. These objects interact with the coordinator class running on the PC which takes as input the suitability measures from each of these camera-agent classes and then allocates the roles accordingly. The camera-agent classes also keep the coordinator class updated about their current performance metric at regular intervals. Thus the coordinator can be sure that the

various tasks are undertaken appropriately and what is the current system performance level. Any time the coordinator class decides that role-swapping is required it can send the newer task allocation commands to the various camera-agent classes. Thus we exploit the physical co-location of all the camera-agent classes to ease the implementation in our current set of experiments. It is worth noting that there may be a number of practical challenges in making such a task allocation/coordination 'truly' distributed. We are looking into this aspect as part of our future work.

However, we intend to use the same generic protocol for performing the coordination amongst multiple sensors. We expect each sensor to send a suitability metric for the system sub-goals at regular intervals. We only want to transfer such meta-data (a few bytes) and not the whole image set etc. Similarly the role allocation commands shall be pre-assigned textual commands which are very small in size. The coordinator can also ask for any helpful parameters from the currently allocated sensor before issuing the role re-allocation commands or passing such helpful parameters to the newly allocated sensor. On the whole, we intend to continue to use the paradigms of meta-level information sharing and process abstraction rather than passing huge image data sets or complicated processing tasks in between the sensors.

Also, please note that we have focused only on single intruder scenarios for our current implementation and experimentation. This has been done in order to circumvent the complexities of multiple object tracking which is indeed a challenging open problem in surveillance research. For example, we currently assume that the location parameters passed between cameras are those of the *only* intruder. This assumption may not hold in multi-intruder scenarios and hence we shall need to verify the intruder identity. This could possibly be achieved by using biometric features or checking the shirt-color or by using a different (non-vision) sensor modality. We hope to explore these issues in our future works.

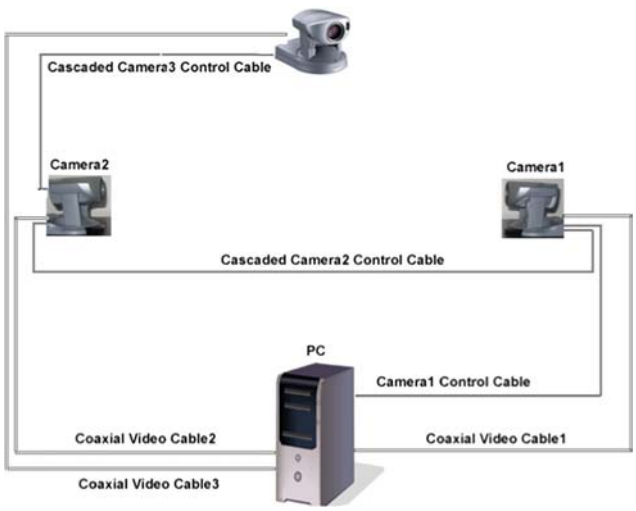

Fig. 9 Three camera system

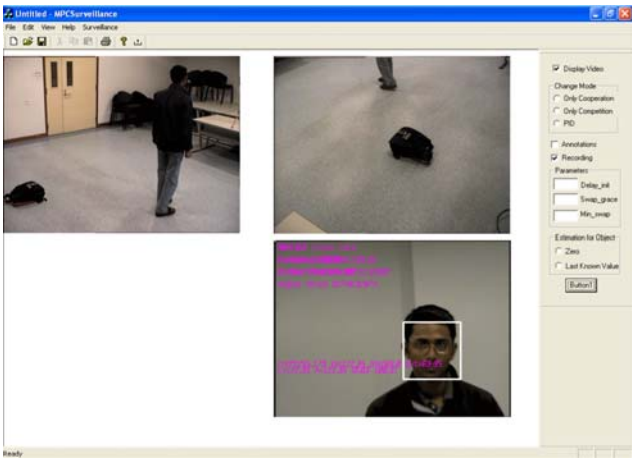

Fig. 10 GUI of the software application

The physical layout for the triple camera setup as it is currently implemented has been shown in Fig. 9. The three cameras were connected in a daisy chain setup as is supported by the Canon cameras. The camera control commands were transmitted using custom-made cable which connected to RS-232C port on the PC. The image data was captured using a Picolo-Pro video capture card with co-axial data cables.

A software interface application was developed for easy monitoring of the system performance as well as setting the parameters like interaction mode, delay in role-exchanging etc. A snapshot of the application GUI is shown in Fig. 10, with the three cameras undertaking their respective tasks.

5 Results and analysis

To establish the veracity of our proposed framework in performing surveillance tasks, we have conducted two sets of comprehensive experiments to compare the *coopetitive* interaction approach and MPC feedback mechanism with their possible alternatives. While one set demonstrates how our *coopetitive* interaction strategy performs against similar dual camera systems, the second set demonstrates the scalability of the *coopetitive* interaction approach to a framework with three cameras which have to undertake three different roles. In both the sets, we compare the *coopetitive* interaction approach with other plausible approaches such as 'only cooperation' and 'only competition'. We also compare the performance of the proposed MPC feedback mechanism with that of the traditional PID control. Besides this, we also conduct a set of experiments to find the appropriate time for role swapping. The idea is to verify the suitability of our proposed random approach for camera to sub-goal allocation as compared to the conventional intruder-follower approaches.

Note that we do not use standard data sets, like those described in PETS [6], to evaluate the performance of our surveillance system, as such data sets do not provide any means for estimating the performance of real time surveillance experiments. They provide off-line images to be used for evaluating the performance, which is not possible in our system as it needs to undertake physical panning, tilting and zooming operations in real time in order to capture the M_{Obj}. Hence, we use real-time experiments to evaluate the performance of our system.

5.1 Dual camera experiments

This section describes the experiments undertaken to authenticate the application of the proposed framework in a dual camera surveillance setup. Our system goal, as discussed earlier, is obtaining frontal facial images of an intruder in a rectangular premises.

We describe briefly the first three experiments conducted in dual-camera setup as follows. Experiment 1 helped us in determining which camera interaction approach or feedback mechanism provides us with best ability to obtain images of intruders traversing certain definitive trajectories in an enclosed environment. In Experiment 2, we determine which interaction approach or feedback mechanism can detect an intruder most number of times in a given time period even if the intruder is allowed freedom to *choose his own trajectory* and *intentionally try to avoid detection*. Experiment 3 provides a comparison between MPC and PID feedback mechanisms in terms of their ability to track and obtain high resolution images of the M_{Obj}. Please note that the experiments and the results for two camera setup have earlier been described in [22] which can also be referred for further details.

Here, we reproduce a summary of the results obtained. As can be seen in Fig. 11, we notice that coopetitive approach along with MPC feedback mechanism has performed significantly better than other approaches. In experiment 1, for three chosen trajectories which varied from simple to complex, we notice that 'Coopetitive with MPC' approach was able to obtain good quality images for over 55% of times. This was better than 'Coopetitive with PID' whose performance began to drop significantly as the trajectory became more complex and 'Only cooperation' and 'Only competition' which performed well below the proposed approach.

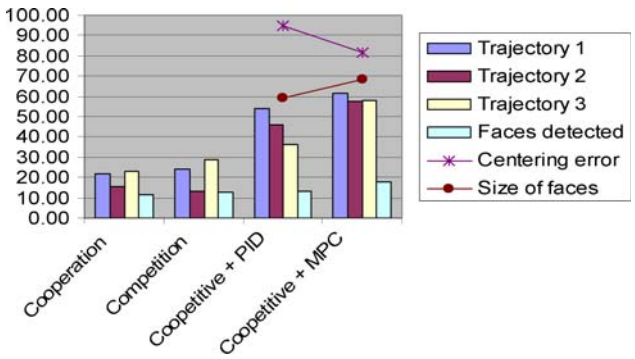

Fig. 11 Comparison summary between different frameworks for the dual camera setup

In experiment 2, we noticed that on average the 'Coopetitive with MPC' approach was able to obtain facial images of an intruder 17.9 times per minute which was greater than 13 for 'Coopetitive with PID' and 12.4 and 11.3 for 'Only Competition' and 'Only Cooperation' respectively.

In third experiment, where we specifically compare 'Coopetitive with MPC' with the 'Coopetitive with PID' approach, we found that the 'Coopetitive with MPC' approach on average obtained 16% less error in centering the M_{Obj} to the center of the image plane and 15% higher resolution for the facial images obtained. These improvements are due to the fact that MPC uses explicit forward estimation to predict the person's position and hence is able to counter the various delays.

Hence, based on these results we were able to conclude that the 'Coopetitive with MPC' interaction approach performed significantly better than the other approaches for the described dual-camera surveillance setup.

5.2 Dynamic role-swapping experimentation

As part of our generic framework, we have highlighted that the system should continually keep checking the appropriateness of the roles assigned to the sensors. If at any time, the over-all system performance can be improved by swapping roles between cameras, then it should be undertaken. Needless, to say that the 'appropriate time' depends on the nature of the task and adversary behavior. As discussed in Sect. 3.6 we want to improve the system performance by adopting a random approach for role swapping.

Besides this we also want to cut down the number of unnecessary role-swaps which may be caused by system implementation noise and irregularities. In our implementation, we employ the conventional 'face detection' for focusing on and obtaining images. As we are aware that mechanisms like face detection are noisy i.e. not 100% accurate, we need to devise some counter mechanisms.

Thus in this subsection we test our two hypothesis to better undertake the dynamic role transfer.

1. Costly operations like role-exchange should not be undertaken based on one (noisy) reading i.e. one negative face detection.
2. A random approach to role swapping can improve the system performance.

For this set of experiments, we considered a scenario where an intruder comes inside the surveyed area, picks up an object and leaves. He can spend minimum 30 s and maximum 1 min inside the surveyed premises. The intruder must try not to get his face captured even a single time. The camera system on the other hand tries to capture his facial images as many times as possible.

For the purpose of these experiments we employed a dual camera setup similar to that described in the preceding subsection. However, while those results were described earlier in [22], the role-swapping experimentation results are being described for the first time in this paper.

We conducted 25 rounds of experiments with each role-exchanging strategy and the average results have been shown in Fig. 12. We define successful detection as obtaining at least one good quality image of the intruder. Hence successful detection percentage reported is the percentage of experiment rounds for which we found at least one good quality image of the intruder. We also report the number of good quality images found on average for each role exchange option.

We find that exchanging roles each time the person's face is not found might not be optimal. This is due to the fact that the person might still be there but the face detection algorithm was not able to detect it. It was found much better to exchange roles if the face was not detected for three consecutive frames, as there is a higher probability that the person *actually* changed his facial direction. We also found that random role-exchanging time with average T_{persist} of 5 s gave better detection percentage and facial image capturing performance than fixed role-exchanging every 5 s. This is because *some* of the intruders were able to estimate the role-exchange time after the first 15 to 20 s in case of the fixed role-exchanging strategy.

Another interesting point to note is that as the T_{persist} increases the probability of capturing the face at least once also increases. But this comes with a trade-off for the number of times the face is captured. Intuitively it makes sense that if the system constantly focuses in one direction then the probability of the person looking at least once in that direction is high. However, the probability that he will continue to look in the same direction is low.

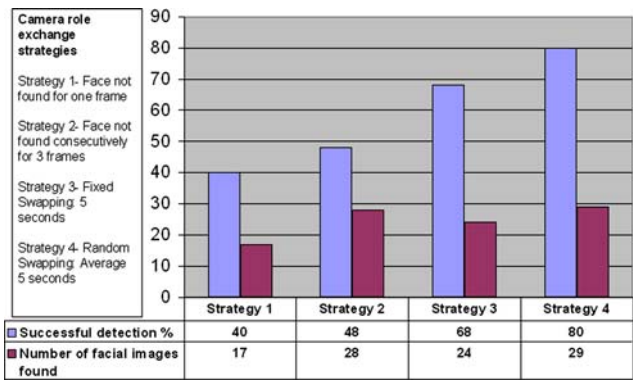

Fig. 12 Comparison summary between different role-exchange strategies

Thus based on this set of experiments we were able to verify that:

1. It is better to carefully check the incoming data before undertaking costly operations like role-swapping
2. A random role swapping strategy does perform better than a fixed time swapping strategy.

5.3 Triple camera experiments

To test the versatility of our proposed 'Coopetitive with MPC' approach, we test its adaptability to a three camera surveillance setup. As discussed earlier, the system should be able to detect any abandonment event and then obtain images of the person leaving the object, the object itself and the trajectory of the person. To understand this process better, let us take a closer look at one such scenario as shown in Fig. 13.

The images from the three cameras have been shown in the figure at various time intervals. We describe them as follows:

1. Set(a) frame 5: The images from the three cameras at the time the person is entering the surveyed premises.
2. Set(b) frame 89: The person is leaving an object.
3. Set(c) frame 149: The 'abandonment event' is detected. Camera1 is allocated the faceCam task. Camera2 is allocated as the objectCam and Camera3 takes over as trajCam.
4. Set(d) frame159: Camera1 is recording trajectory. Camera2 has focused on the object. Camera3 is trying to focus on the face.
5. Set(e) frame 175: Camera2 has focused and zoomed on to the face
6. Set(f) frame 240: Person leaving the surveyed premises.
7. Set(g) frame 350: Camera2 zooms further into the object to obtain to higher resolution object images.

We notice from this scenario shown in Fig. 14 that the appropriate images have been captured by the cameras based on the correct role allocation for them. In fact, the interaction mechanism used for this example scenario was 'coopetition'. One point to note is that camera3 has correctly been allocated as the faceCam. This is because it is the only camera that has been able to obtain frontal facial images amongst the three cameras, and hence the correct allocation allows us to capture facial images appropriately. Another important point to note is that camera2 was able to focus on to the object even though it did not even see the object on its own. Based on the parameters passed by the eventCam however it was able to focus on to the object.

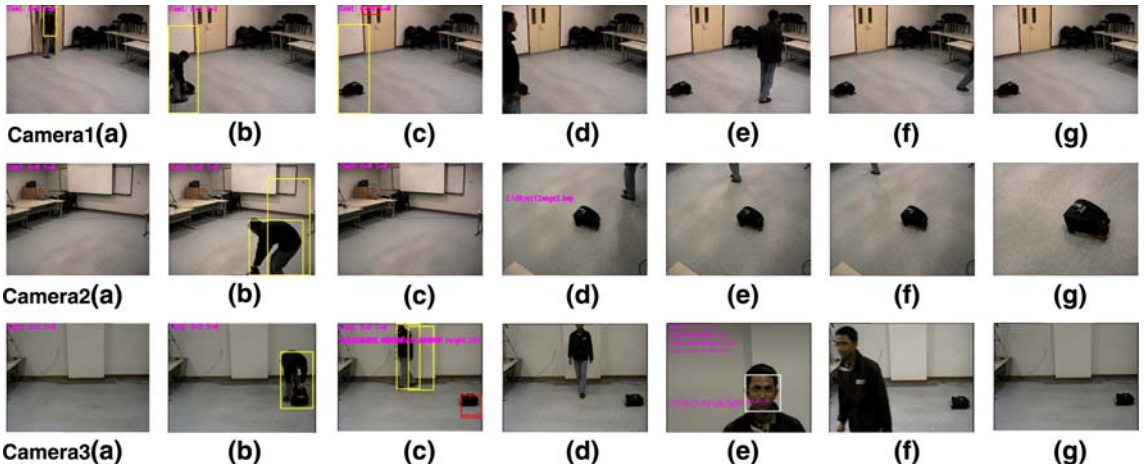

Fig. 13 One Scenario

Now that we have a clear idea about the system goal and sub-goals it is trying to undertake, let us discuss further the experimental results obtained for the different interaction strategies and feedback mechanisms in such a surveillance setup.

For the purpose of this set of experiments, we define 'only cooperation' as the interaction strategy in which each sensor is assumed to be fully capable of handling any given system task. Hence the parameters as well as roles are freely passed to all cameras by the camera which detects the abandonment activity. The roles are allocated by a random distribution method.

The 'only competition' mode on the other hand, assumes the cameras to be separate entities trying to obtain the most coveted task and then performing the allocated task as well as possible without any help from the other sensors. This interaction mechanism follows a strict measure of merit to assign the tasks to the various cameras. The facial image capturing which is the most restrictive is allocated to only one of the three competing cameras. This camera is decided to be the one which detect frontal face with highest size at the time of event detection. This is followed by the competition between the remaining two cameras for the task of focusing on object for which the measure of merit is the distance between the object and the person. This prevents any occlusion of the object by the person in the camera which gets allocated this task.

The proposed coopetitive interaction mode makes use of the best features of the above mentioned two approaches to obtain better system results. It uses the same measures of merit as 'only competition' to allocate the tasks to the most suitable cameras. It also adopts the parameter passing paradigm of the 'only cooperation' method to pass the correct object location information to the camera allocated the task.

For each of the different interaction strategies, we repeated the experiment 20 times with different object positions. The first important step in successful execution of the surveillance task is the proper allocation of roles between sensors. In particular the allocation of the appropriate camera to the faceCam is of prime importance.

A summary of the faceCam role allocation using the various interaction strategies has been shown in Fig. 14. As can be seen from the figure, we notice that the faceCam role allocation for the 'only cooperation' strategy has not been accurate. This is due to the fact that in 'only cooperation' strategy all camera are assumed to be fully capable of handling the various system tasks. Hence the role allocation is random. This results in the correct camera being allocated the faceCam task only around 35% of the times, which corroborates well with the fact that the role allocation is random. The faceCam role allocation is done correctly for a high percentage of cases (over 80%) for the 'only competition' and both the variants of 'coopetition' strategy. This is due to the fact that in these cases, the role allocation is based on an explicit notion of merit or worthiness between sensors. This means that only a camera which can see the person's frontal face clearly has been allocated the faceCam task.

After the role allocation, let us look into the three system sub-goals. Also, please note that for ease of presentation, we use the average values obtained from the twenty experimental rounds for each strategy in the remaining results. The first system task was to obtain three 'good quality' images of the person who is abandoning the object. This task is in fact closely related to the process of appropriate camera allocation. If an inappropriate camera is allocated this task it would be impossible for it to obtain frontal facial data and hence good quality facial images.

As can be seen from the results shown in Fig. 15, the 'only cooperation' framework was able to obtain high quality facial images only 30% of the times. Again, this corroborates well with the fact that the role allocation is random in 'only cooperation' interaction strategy.

Mode:	Only Cooperation			
Camera allocated as Face cam	Total	Correct	Incorrect	Correct %
1	7	3	4	43%
2	6	2	4	33%
3	7	2	5	29%
Total for 3 cameras	20	7	13	35%

Mode:	Coopetitive (with PID)			
Camera allocated as Face cam	Total	Correct	Incorrect	Correct %
1	7	6	1	86%
2	6	6	0	100%
3	7	6	1	86%
Total for 3 cameras	20	18	2	90%

Mode:	Only Competition			
Camera allocated as Face cam	Total	Correct	Incorrect	Correct %
1	7	6	1	86%
2	6	6	0	100%
3	7	7	0	100%
Total for 3 cameras	20	19	1	95%

Mode:	Coopetitive + MPC			
Camera allocated as Face cam	Total	Correct	Incorrect	Correct %
1	7	7	0	100%
2	6	5	1	83%
3	7	6	1	86%
Total for 3 cameras	20	18	2	90%

Fig. 14 FaceCam role allocation for the three cameras

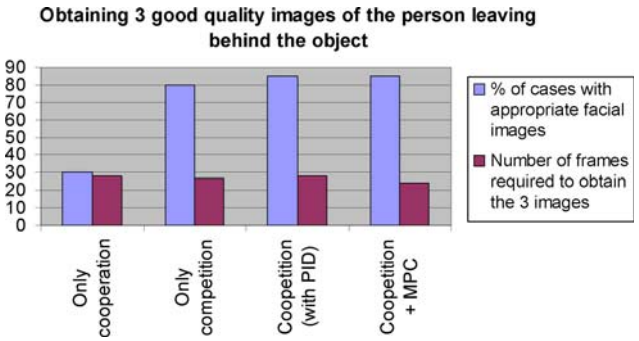

Fig. 15 Comparison between different frameworks for obtaining three good quality images of the person leaving behind the object

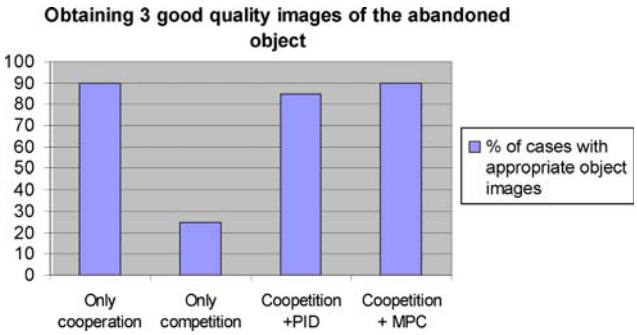

Fig. 16 Comparison between different frameworks for obtaining three good quality images of the abandoned object

On the other hand, the 'only competition' and 'coopetition' strategies both employ a suitability yard stick to allocate the face capturing task. Hence we notice a high percentage (over 80%) success rate for these two interaction strategies.

Another important yard-stick for measuring performance is to see how fast the three good quality images are captured. This is important as the intruder tries to get away from the scene as possible. We noticed it took 28 frames on average for the system to capture three facial images if the PID feedback was used. However, if we change the feedback mechanism to MPC we are able to obtain the images in only 24 frames on average. This signifies a 19% improvement in the speed of obtaining three high quality images. The improved performance with MPC can be explained due to the fact that it uses a forward state estimation to track the person better.

The second system task is to obtain high quality images of the abandoned object. The correct position parameters of the object need to be passed from the camera detecting the 'abandonment' event to the most suitable camera.

As can be observed from Fig. 16, the only competition strategy was able to obtain images good quality images only in 25% of the cases. This was due to the fact that in only competition strategy, no object position parameters are passed from one camera to the another. Hence, the only scenario in which a camera may obtain appropriate object images is that in which the object and the person were also visible to this camera at the time of event detection and hence it can make a correct object position inference on its own. The only cooperation and coopetitive interaction strategies are able to obtain images for over 85% of the cases as they obtain the appropriate position information from the event detection camera.

The third system task is to obtain the correct trajectory information for the person abandoning the object. This task does not need active movement of the camera but is still important as an evidence of the over-all happenings in the scene as well as the trajectory of the person. For example the fact that a person exited from a window instead of the usual exit door can significantly increase the suspiciousness of the object he has left behind.

As can be seen in Fig. 17, we were able to obtain the images of the person's trajectory in the surveyed premises for over 90% of the cases. We are not able to obtain person's trajectory in all cases as the camera field-of-view was not able to cover the entire surveyed area. As this part requires static camera placement rather than any active sensing, we notice that all the different interaction strategies perform almost equally well.

After looking at the results piece-meal for each of the three system tasks let us now undertake an overall comparison of the results obtained for the the three system tasks. The summarized results have been shown in Fig. 18. We realize that coopetitive interaction strategy combined with MPC feedback has consistently performed well in all the tasks. This framework has significantly outperformed only cooperation strategy in terms of the capability to obtain facial images of the person abandoning the object. Similarly, it has also severely out-performed only competition strategy in terms

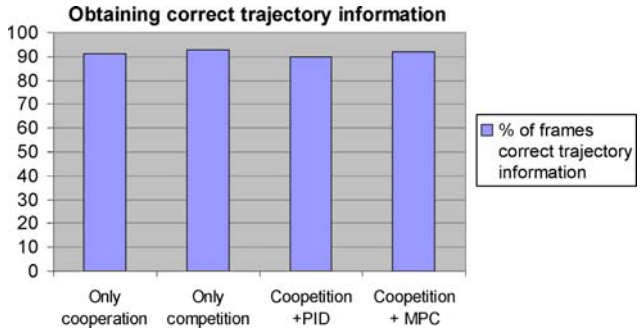

Fig. 17 Capturing the trajectory of the person abandoning the object

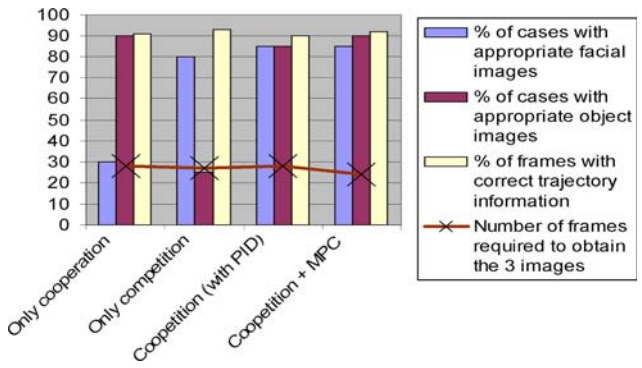

Fig. 18 Summary of the results

of obtaining good quality object images. Lastly, the MPC feedback mechanism has provided better tracking capability to the cameras and hence we see that the number of frames required to obtain three good quality images of the person has been reduced by 19% as compared to the PID feedback mechanism.

Hence, this set of experimentation has re-affirmed to us that coopetitive interaction strategy can provide significantly better results than other plausible alternatives.

5.4 Further discussion

While undertaking the practical implementation of our proposed systems and following experimentation, we faced a number of practical challenges. Such issues may not have a significant theoretical merit but are indeed an important aspect of practical implementation.

We feel that such issues are often faced but rarely documented in research works. Hence, we would like to document a summary of the practical challenges faced in the course of our experimentation. Some of the key challenges faced and our adopted counter-actions were as follows:

1. Camera control: We used a daisy chain of Canon VC-C4 cameras connected to a PC to pass the camera control commands. We faced a problem that the some of the commands given to cameras were not being implemented. Later, we realized that if multiple commands are passed consecutively then only the last given command is actually transferred and the other commands get aborted. Hence we need to provide a delay between various control commands to allow the commands to be passed to the cameras correctly.
2. Face and blob detection: Our system uses blob detection and face detection for the various system tasks, hence its accuracy is important for our framework to work well. We faced some issues in detecting correct blobs and faces. We were able to improve the performance to a higher accuracy by using ambient lighting and adjustment of various parameters. Also, for face detection we adopted the policy of three consecutive detections to conclude on presence or absence of faces.
3. Real time experimentation: In our experiment we were dealing with real-time camera controlling, hence we needed to set-up the complete system before each debugging/experimentation. As we were using ad-hoc setting up of the equipment, this involved significant set-up costs each time we conducted experiments. In future, we are considering creating a permanent customizable test-bed for undertaking debugging before performing actual real-time experiments.

To sum up our discussion for this section, we have proposed a generic active multi-sensor framework with *coopetitive* interaction and MPC. We have conducted experiments with two different adaptations of the framework. We have also conducted a preliminary study into finding appropriate time for exchanging roles between sensors.

We found that 'coopetitive' interaction approach was better able to perform the system tasks than the 'only cooperation' strategy for the two dual camera experiments as well as the facial image capturing in the the triple camera experiment. It also performed better than the 'only competition' interaction strategy in the two dual camera experiments as well as for obtaining object images in the triple camera experiment.

We also found that a random approach to role-exchange between sensor can push the 'burden of out-guessing' on to the adversary and help in improving the system performance.

We also found that MPC contributed significantly toward improving system performance. In the two camera adaptation, it performed significantly better (56% frames with appropriate images) as compared to the PID approach (36% of frames) for frontal image capturing task in a complicated trajectory. Similarly the intruder detection rate was 27.4% higher using MPC as compared to PID. Also, the average centering error was reduced by 16.5% and average facial image size increased by 13.2% by using MPC instead of PID. The positive trend by adoption for MPC has continued in the triple camera scenario where the time taken to capture

three facial images was reduced by 19% with the use of MPC. Each of these improvements can significantly alter the overall system impact in a sensitive application like surveillance where a 15–20% impovement in capturing speed or image size could prove to be the difference between a criminal being identified or going unnoticed. It is also worth noting that MPC has consistenly outperformed PID and these improvements have been more marked in complex scenarios and scenarios where there needs to be frequent parameter/role transfer between multiple sensors.

Hence, based on our variety of experiments conducted with both dual and triple camera set-ups, we are able to conclude that a 'coopetitive with MPC' based framework does indeed perform significantly better than other comparable strategies and MPC does out perform the conventional PID feedback strategy.

6 Conclusions

In this paper we have proposed an enhanced generic framework for multi-sensor environments which has novel features in terms of its mode of interaction and feedback mechanism. We have proposed *coopetitive* approach for interaction between sensors which allows sensors to cooperate based on a merit decided by competition. We have also established the value of MPC as an efficient feedback mechanism that can help to counterbalance various transfer and reaction delays observed in cooperative sensing.

We have also tested the adaptability and the scalability of the proposed framework via two different surveillance adaptations which differ in terms of their surveillance goal as well as the number of sensors employed. While the dual camera adaptation aimed at capturing intruder faces as many times as possible in an enclosed environment, the triple camera setup aimed at obtaining detecting 'abandonment' events followed by getting images of the person abandoning the object, the object itself and the trajectory undertaken.

From the results for the both dual and triple camera adaptations, we can clearly conclude that for interaction between sensors, *coopetition* i.e. cooperation based on merit performs significantly better than 'only cooperation' or 'only competition' approaches. We also deduce that MPC performs significantly better than PID as a feedback mechanism for vision sensors. This is by virtue of MPC's capability to consider estimated future values rather than just past data to make its control decisions.

Future work scope in this area remains in creating more precise means for defining and handling cooperation and competition between sensors. More sophisticated means of estimating future states would also be very useful.

We are currently working on extending the proposed framework to handle multiple intruders and multiple sensors which could be visual as well as non-visual e.g. audio sensor, infra-red sensors etc. Such non-visual sensors would allow the framework to handle situations where visual sensing alone might fail e.g. intruder hiding the face or using a face-mask etc. Besides this, we are also exploring how we can make the sensing system truly distributed where the sensors are not coordinated by a central agent.

References

1. Barreto, J.P., Batista, J., Peixoto, P., Araujo, H.: Integrating vision and control to achieve high perfomance active tracking. Tech. rep., TR-BAR-0202, ISR/DEEC, University of Coimbra (2002)
2. Bowyer, R.S., Bogner, R.: Cooperative behaviour in multi-sensor systems. In: International Conference on Neural Information Processing, pp. 155–160. Perth, Australia (1999)
3. Castanedo, F., Patricio, M.A., Garca, J., Molina, J.M.: Extending surveillance systems capabilities using BDI cooperative sensor agents. In: ACM International Workshop on Video Surveillance, pp. 131–138. Santa Barbara, USA (2006)
4. Collins, R., Lipton, A., Fujiyoshi, H., Kanade, T.: Algorithms for cooperative multisensor surveillance. Proc. IEEE **89**(10), 1456–1477 (2001)
5. Collins, R., Lipton, A., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O.: A system for video surveillance and monitoring. Tech. rep., CMU-RI-TR-00-12, Robotics Institute, CMU, USA (2000)
6. Datasets: In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. Breckenridge, USA (2005)
7. Dias, M., Stentz, A.: A market approach to multirobot coordination. Tech. rep., CMU-RI -TR-01-26, Carnegie Mellon University (2001)
8. Garcia, J., Carbo, J., Molina, J.M.: Agent-based coordination of cameras. Int. J. Comput. Sci. Appl. **2**(1), 33–37 (2005)
9. Greiffenhagen, M., Ramesh, V., Comaniciu, D.: Statistical modeling and performance characterization of a real-time dual camera surveillance system. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 2335–2342. Hilton Head Island, USA (2000)
10. Khadir, M., Ringwood, J.: Linear and nonlinear model predictive control design for a milk pasteurization plant. J. Control Intell. Systems **31**(1), 37–44 (2003)
11. Lam, K.Y., Chiu, C.K.H.: Adaptive visual object surveillance with continuously moving panning camera. In: ACM International Workshop on Video Surveillance and Sensor Networks, pp. 29–38. New York, USA (2004)
12. Laurent, D., Mustapha, E., Claude, P., Pascal, V.: A mobile robot localization based on a multisensor cooperation approach. In: IEEE International Conference on Industrial Electronics, Control, and Instrumentation, pp. 155–160. New York, USA (1996)
13. Liu, Q., Kimber, D., Foote, J., Wilcox, L., Boreczky, J.: FLYSPEC: a multi-user video camera system with hybrid human and automatic control. In: ACM Conference on Multimedia, pp. 484–492. New York, USA (2002)
14. Marchesotti, L., Messina, A., Marcenaro, L., Regazzoni, C.: A cooperative multisensor system for face detection in video surveillance applications. Acta Autom Sinica Chinese J. Autom. **29**, 423–433 (2003)
15. Molina, J., Garca, J., Jimenez, F., Casar, J.: Surveillance multisensor management with fuzzy evaluation of sensor task priorities. Eng. Appl. Artif. Intell. **15**(6), 511–528 (2002)

16. Molina, J., Garcia, J., Jimenez, F., Casar, J.: Cooperative management of a net of intelligent surveillance agent sensors. Int. J. Intell. Systems **1**(8), 279–307 (2003)
17. Morari, M., Lee, J.H., Garcia, C.E., Prett, D.M.: Model Predictive Control. Prentice Hall, Englewood Cliffs (2003)
18. Papanikolopoulos, N., Khosla, P., Kanade, T.: Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. IEEE Trans. Robot. Autom. **9**(1), 14–35 (1993)
19. Roy, P.K., Mann, G., Hawlader, B.C., Masek, V., Young, S.O.: Comparative study of model predictive and decoupled PID controller for a multivariable soil heating process. In: IEEE Newfoundland Electrical and Computer Engineering Conference. Newfoundland, Canada (2004)
20. Saedan, M., Jr, M.H.A.: 3D vision-based control on an industrial robot. In: IASTED International Conference on Robotics and Applications, pp. 152–157. Clearwater, USA (2001)
21. Sharkey, P., Murray, D.: Delays versus performance of visually guided systems. IEE Proc. Control Theory Appl. **143**(5), 436–447 (1996)
22. Singh, V.K., Atrey, P.K.: Coopetitive visual surveillance using model predictive control. In: ACM International Workshop on Video Surveillance and Sensor Networks, pp. 149–158. Singapore (2005)
23. Swarup, S., Oezer, T., Ray, S.R., Anastasio, T.J.: A self-aiming camera based on neurophysical principles. In: The International Joint Conference on Neural Networks, pp. 3201–3206. Portland, USA (2003)
24. Wang, J., Zhang, C., Shum, H.: Face image resolution versus face recognition performance based on two global methods. In: Asian Conference on Computer Vision. Jeju Island, Korea (2004)
25. Zhou, X., Collins, R., Kanade, T., Metes, P.: A master-slave system to acquire biometric imagery of humans at distance. In: ACM International Workshop on Video Surveillance, pp. 113–120. Berkley, USA (2003)

Author biographies

Vivek Kumar Singh is currently working as a Research Assistant at the National University of Singapore. He has obtained B.Eng (Comp. Eng.) and Master of Computing (part-time) degrees from the same university in years 2002 and 2006 respectively. He also worked as a Lecturer at the Institute of Technical Education, Singapore from July 2002 to April 2006. His research interests lie in Multimedia Surveillance and Active Media Sensing.

Pradeep Kumar Atrey is a Research Fellow at the Multimedia Communications Research Laboratory, University of Ottawa, Canada since August 2006. At the time of writing this article, he was with School of Computing, National University of Singapore, where he obtained his Ph.D. in Computer Science in July 2006. He has also worked as a lecturer at Delhi College of Engineering, University of Delhi and at Deenbandhu Chhotu Ram University of Science and Technology, Murthal (Haryana), India from 1992 to 2002. His research interest includes Video/Audio Processing, Assimilation and Analysis, Multimedia Surveillance, Event Detection, and Multimedia Security.

Mohan Kankanhalli obtained his BTech (Electrical Engineering) from the Indian Institute of Technology, Kharagpur and his MS/Ph.D. (Computer and Systems Engineering) from the Rensselaer Polytechnic Institute. He is a Professor at the School of Computing at the National University of Singapore. He is on the editorial boards of several journals including the ACM Transactions on Multimedia Computing, Communications, and Applications, IEEE Transactions on Multimedia, ACM/Springer Multimedia Systems Journal, Pattern Recognition Journal and the IEEE Transactions on Information Forensics and Security. His current research interests are in Multimedia Systems (content processing, retrieval) and Multimedia Security (surveillance, authentication and digital rights management).