Local Reconfiguration for Simultaneous Coverage and Tracking in a Large Scale Camera Network

Bryan Lemon, Vinod Kulathumani
Lane Dept. of Computer Science & Electrical Engineering
West Virginia University
bryan@bryanlemon.com, vinod.kulathumani@mail.wvu.edu

***Abstract*—**
We present a distributed camera control algorithm that can be used for the monitoring of large regions using a network of PTZ (pan/tilt/zoom) cameras. The proposed strategy would allow for continuous target tracking at a high resolution, while still maintaining coverage over the entire monitored region at the highest possible resolution. Our algorithm requires only a local exchange of information, is quick to converge and ensures longer periods of stability between successive reconfiguration steps. We evaluate the performance of our algorithm in simulation and demonstrate that local reconfiguration is sufficient for maintaining an acceptable coverage of target and non-target points.

I. Introduction

Networks of actively controllable pan / tilt / zoom (PTZ) cameras are increasingly being deployed for surveillance purposes in urban environments. In this paper, we consider a large scale network of PTZ cameras that can be used for monitoring and protecting critical assets such as our national border and civil infrastructure like bridges, highways and airports. The design of such large scale camera systems poses several challenges. First of all, given that the system can span potentially several kilometers, the cameras have to be deployed to ensure optimal coverage with minimum number of cameras. Secondly, while it is required that targets of interest are continuously tracked, it is equally important to ensure that the remaining cameras reconfigure themselves to cover all non-target points. Thirdly, it is important that the cameras converge to new configuration parameters quickly in the presence of mobile targets without a network-wide exchange of information.

To address these challenges, in this paper we describe a local reconfiguration algorithm for surveillance of targets in a large scale network of actively controllable PTZ cameras. The following are key features of our technique: (1) In contrast to algorithms that focus only on continuous target tracking, our algorithm ensures that targets in the scene are tracked at a desired resolution while at the same time ensuring that non-target points in the network are covered at the highest possible resolution. By doing so, we simultaneously ensure that the perimeter is not breached and that there are no voids of coverage when certain cameras are occupied with tracking of targets. (2) We avoid iterative techniques for ensuring coverage where individual cameras make several rounds of adjustments to their PTZ settings based on local message exchanges. Instead, in our algorithm, cameras converge to a configuration decision in a single round of message exchange. Furthermore, information is exchanged only within a local neighborhood as opposed to being global or network-wide. By being local and non-iterative, we ensure quick convergence in an ever changing environment. (3) Instead of triggering a camera reconfiguration every time a target moves, we resort to reconfiguring the system only when a target has moved beyond the range of the camera that is currently tracking the target. This allows the system to be in a stable state for longer periods of time as opposed to almost always being in a reconfiguration phase to keep pace with a target. Longer stable periods also reduce the burden on the underlying network by reducing the required data transfer rates.

We provide an extensive evaluation of the performance of our algorithm by analyzing the impact of network density and the size of the locality of stabilization regions on the achievable coverage resolution. We demonstrate that we are able to track targets at a high resolution, observe the entire space at an acceptable resolution, and keep the observing cameras stable for long periods of time by only performing the reconfiguration in a local area around the target.

Related work: A significant amount of research has been carried out on automated tracking of targets using a camera network. Some of these efforts have focused on the design of multi-camera systems to automatically detect, segment, and track targets [1]–[3]. Others have focused on algorithms for efficient handoff of targets in multi-camera multi-target surveillance scenarios [4]–[6]. However, there has been relatively less work on multi-camera systems that track targets at high resolution while simultaneously ensuring that other areas are completely covered.

In [7], a decentralized algorithm is proposed to maximize the coverage of a region monitored by PTZ cameras by ensuring there are minimum overlaps with neighboring cameras. However, by seeking a global optimization and by being iterative in nature the algorithm takes a long time (several seconds) to converge making it less suitable for dynamic adjustments when targets are being tracked. By way of contrast, we focus on a local and non-iterative solution in which each camera involved in the reconfiguration makes only one adjustment of its PTZ setting to reach the new configuration. We also note that the work in [7] focused on maximizing coverage of an area while we additionally focus on maximizing the resolution at which an area is covered.

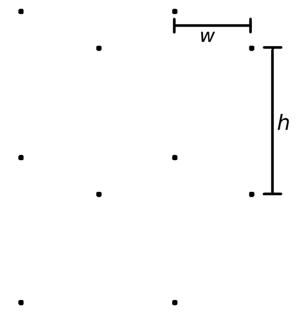

Fig. 1. The layout of the cameras as visualized on a plane parallel to the ground plane at a height H above the ground.

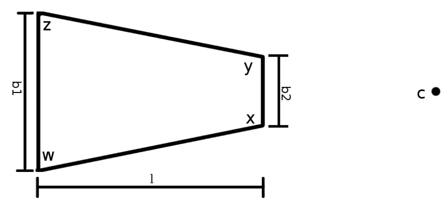

Fig. 2. A projection of the FOV on the ground plane for a camera c outlined by points $wxyz$.

In [8], an algorithm is presented to provide optimal coverage over an area using downward facing hovering cameras (that are mobile). This algorithms also seeks global optimal coverage and is not intended for fast dynamic reconfiguration as considered in this paper. Also, in this paper we have considered cameras which are not downward facing, are more practical to deploy for monitoring large regions and have 2 additional degrees of freedom: panning and tilting.

Outline of the paper: In Section 2, we present the system model. In Section 3, we describe our dynamic reconfiguration algorithm. In Section 4, we describe our experimental design and evaluate the performance of our algorithm. In Section 5, we conclude and provide directions for future work.

II. System Model

In this section, we describe the layout of the cameras and the parameters that the cameras control to affect coverage. We also define various system parameters, describe the perturbation model and then formally state the problem.

A. Camera deployment

We consider a network of N_c PTZ cameras deployed to monitor a region with area A_M. Let P denote the resolution of each camera which is equal to the total number of pixels in each frame captured by the camera. The cameras are deployed at a uniform height H above the ground. Consider the horizontal plane parallel to the ground plane at a height H: On this plane, the camera deployment can be visualized as an uniform interlocked grid as shown in Fig. 1. Let w denote the separation between any two cameras along a row and let h denote the separation between two rows as depicted in Fig. 1.

The cameras have three knobs for controlling the coverage of the region: pan, tilt and zoom. The cameras are assumed to be able to pan across the full $360°$. The cameras can tilt between a minimum of $35°$ and a maximum tilt of $45°$ declination from the horizon. The cameras are also assumed to have a $5X$ optical zoom capability. Note that instead of downward facing hovering cameras (that lend analytical simplicity to the problem by projecting into the ground plane as squares or circles), we have assumed a more practical deployment scenario. Because of the assumed tilt parameters, the projection of the field of view of each camera into the ground plane can only be modeled as a trapezoidal shape. Fig. 2 shows an example of the projection of a camera's field of views on the ground plane in an area defined by points $\{$*w, x, y, z*$\}$. In describing our algorithm, We refer to edges wx and yz as the side edges of a camera's field of view (FOV) and refer to edge wz as the leading edge of a camera's FOV. From hereon, we implicitly refer to a camera's FOV as its field of view on the ground plane.

B. System parameters

Definition 2.1 (Coverage resolution offered by camera c): At a given PTZ setting, if A_c is the area monitored by the FOV of camera c, then the coverage resolution offered by camera c (denoted as R_c) is $\frac{P}{A_c}$.

Thus, the greater the area monitored by a camera at a given setting, the lower the coverage resolution offered by that camera. Tilting up and thus moving away from the ground plane causes the coverage resolution offered by a camera to decrease. Likewise, zooming out also decreases the coverage resolution offered by a camera.

Definition 2.2 (Coverage resolution of point p): Let $\{C_p\}$ denote the set of cameras whose FOVs cover point p. The coverage resolution of point p (denoted as R_p) is $\max_{c\in\{C_p\}} R_C$, i.e., the maximum of the coverage resolution offered by all cameras in $\{C_p\}$.

Definition 2.3 (Coverage resolution of area A): Coverage resolution of area A (denoted as $R(A)$) is $\min_{p\in A} R_p$, i.e., the minimum coverage resolution among all points in area A.

Note that if a point is not covered by any camera, it has zero resolution. Area A is said to be *completely covered* if each point in A has non-zero coverage resolution.

C. Initial state

For the camera network, we define a special state called the *initial* state. This corresponds to the unperturbed state for the system when no targets are present and all cameras are in their default configuration. In this state, the PTZ setting for each camera is adjusted such that each point is covered at the highest resolution possible for the given deployment. The default configurations for the initial state are determined offline and set during system deployment. For the symmetric deployment described above, the initial state is determined as alternate rows of cameras panned to face each other and the tilt/zoom parameters adjusted such that there is no overlap in coverage between adjacent cameras. This gives rise to an interlocked pattern as shown in Fig. 3. An algorithm such as [7] could also be used to determine this initial configuration in

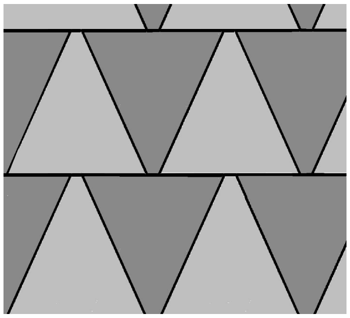

Fig. 3. The *initial* state for the camera network. Cameras in alternate rows are panned to face each other. All points are covered at the highest resolution, with no overlap between the cameras' FOVs.

N_c	Number of cameras in the system
$\{C\}$	The set of all cameras in the network
c_{le}	The leading edge of camera c
c_{se}	The side edges of camera c
c_{pan}	The current pan setting of camera c
c_{tilt}	The current tilt setting of camera c
c_{zoom}	The current zoom setting of camera c
g	Gap: a continuous region of unobserved space
$\{G\}$	The set of all gaps within a local area
$\overline{G}$	The gap with the largest area
A_M	The observed area
A_G	The total area of $\{G\}$
A_c	The area Camera c is currently observing
$x\|\|y$	x and y are adjacent areas with their perimeter touching
$mid(x)$	The centroid of area x

TABLE I
TERMS AND NOTATIONS USED IN THIS PAPER

a self-organizing manner as convergence time is not an issue during initial configuration. Let $R_I(A_M)$ denote the coverage resolution of area A_M in the initial state. We use this initial resolution as a reference to analyze the performance of our algorithm.

D. Perturbation model and problem statement

The following events can cause the camera network system to be perturbed causing reconfiguration of camera parameters: (1) a target enters the system and is assigned to a camera for tracking and (2) a target is reassigned to a different camera for tracking. Each time a perturbation event occurs, the reconfiguration algorithm is invoked at the camera that is assigned for tracking the target. The reconfiguration algorithm determines new configuration parameters for all cameras lying within a local area A_L defined by a circle of radius zh around the target. The parameter z determines the size of the locality over which the reconfiguration algorithm is executed. The goal of the system is then to maximize $R(A_m)$ by reconfiguring cameras that lie within A_L. The effect of parameter z on $R(A_M)$ is analyzed in §IV.

III. ALGORITHM DESIGN

In this section, we describe our distributed control algorithm to ensure complete coverage of area A_M at the highest possible resolution. Each time that the tracking camera is changed, this camera uses the location of all cameras within an area A_L around itself and runs a reconfiguration algorithm. The new

```
while {G} ≠ ∅ do
  for all c in C_L do
    a_before = A_G
    for all G_i in G do
      if c_le||G_i then
        c_tilt += Δtilt
        break
      end if
    end for
    for all G_i in G do
      if c_se||G_i then
        c_zoom −= Δzoom
        break
      end if
    end for
    c_pan moves towards mid(G̅) by Δpan
    a_after = A_G
    if a_before − a_after < 1sqft. then
      undo Δpan
    end if
  end for
end while
```

Fig. 4. Pseudocode for the reconfiguration algorithm

pan, tilt and zoom settings for all cameras within area A_L are determined by the tracking camera and the new parameters are distributed back to all those cameras. When calculating this new configuration, the camera that is assigned for tracking removes itself from consideration and as a result the new configuration stays *valid* for the duration that the tracking camera does not change. We describe our algorithm in three parts: (1) camera assignment for tracking, (2) actively tracking a target and (3) determining reconfiguration parameters.

1) Determining the Tracking Camera: When a new target is detected, or when a target is outside the FOV of a currently tracking camera, each camera within a circle of area A_L around the target is informed of this new event along with the location of the target. In this paper, we use the proximity of a camera to the target to be the tracking camera selection criteria, by which the tracking camera is the camera closest to the target. Each camera within A_L locally computes the Euclidean distance of the target from all cameras within area A_L around the target, and decides if it will be the tracking camera.

2) Tracking the Target: The camera assigned for tracking pans, tilts, and zooms to keep the target in the center of its field of view while maintaining the highest resolution possible. When the camera is unable to track the target at a high resolution, the system finds an alternative camera to do the tracking as described above.

3) Reconfiguration: The pseudo-code for the reconfiguration algorithm is provided in Fig. 4. The reconfiguration algorithm is run on-board the tracking camera. The location of each camera within area A_L around the target is provided as input to the reconfiguration algorithm.

Note that if the goal was simply to cover the gap that is created due to the loss of the tracking camera for actively following a target, then one or more cameras whose FOV is adjacent to the gap could simply tilt up and zoom out to cover the gap. However, our focus is also in minimizing the loss of coverage resolution. Hence in our approach, the cameras also pan towards the centroid of the gap so that the loss of resolution is shared by cameras farther away from the gap. For a globally optimal solution, the parameters of cameras through out the network would need to be adjusted to minimize the loss of resolution but this is infeasible in a large network. Hence only the PTZ parameters of cameras within A_L are adjusted. In §IV, we test the impact of different sizes of A_L on the loss of resolution.

In order to determine the new configuration, the cameras within A_L are first ordered based on proximity to the location of the tracking camera. Let this ordered set of cameras be denoted as $\{C_L\}$. The initial configuration for all these cameras is marked as the configuration corresponding to the *initial* state which is described in §II. Then, in the predetermined order the PTZ parameters of each camera are incrementally updated at the tracking camera that is computing the new configuration, based on the following steps. (Recall from Table I that each contiguous unobserved area in the network is called a *gap*.)

- R1: The total unobserved area is determined by summing the areas of all *gaps*.
- R2: If the leading edge of a camera's field of view is adjacent to at least one *gap*, the camera's tilt is adjusted so that the length of the camera's FOV (represented as parameter l in Fig. 2) increases by 1 foot. This adjustment is denoted as $\Delta tilt$ in Fig. 4.
- R3: If either of the two side edges of the camera's field of view is adjacent to at least one *gap*, the camera's zoom is adjusted so that parameter $b1$ of the camera's FOV is incremented by 1 foot. This adjustment is denoted as $\Delta zoom$ in Fig. 4.
- R4: The largest *gap* after taking into account the above parameter changes is then determined. The camera's pan parameter is adjusted to move towards the centroid of the largest gap by a unit of Δpan. The unit of Δpan is set to a constant of $3°$. The total unobserved area is then determined by summing the areas of all the gaps after the panning adjustment. If the total unobserved area does not decrease by at least by 1 square foot from the area determined in step 1, the panning adjustment is reverted.

The above steps are repeated for each camera in the predetermined order repeatedly until there are no more *gaps*. The algorithm is guaranteed to converge because each round of the algorithm for a given camera results in reducing the total unobserved area by at least 1 square foot. Note that, the tilting up and zooming out rules (rules 2 and 3) will always result in reduction of unobserved area. The only rule likely to result in an increase in the unobserved area is rule 4, but by performing a convergence check we ensure that this does not occur. Note that the panning step is important to ensure that the loss of coverage resolution is shared by cameras beyond the perimeter of a gap created by removal of the tracking camera. In other words, the cameras adjacent to the newly created gap are able move some of the unobserved area into the vicinity of cameras farther from the gap. As a result the cameras have to tilt and zoom out lesser and this results in reduction in the loss of coverage resolution.

Once the new parameters are determined, they are distributed to all cameras within A_L, which then make a single adjustment directly to the new configuration. When the target leaves the FOV of the currently tracking camera and a new reconfiguration event is triggered, the cameras that are not involved in the new reconfiguration return to their initial state.

IV. Performance Evaluation

In this section, we evaluate the performance of our algorithm in simulations.

A. Simulation model

We simulate an area of 1 Square Kilometer and deploy cameras at different densities to monitor this region. A standard way to define camera density would be the number of cameras per unit area. But this is not meaningful with PTZ cameras that have different coverage resolutions at different settings of tilt and zoom. Hence in this paper, we analyze the impact of network density by measuring density in relation to a reference deployment. Consider the FOV as depicted in Fig. 2, for a camera which is tilted up and zoomed out completely such that the area covered by the camera is maximized and let us denote this as $F\hat{O}V$. We define unit network density to be a deployment with the value of h set equal to l in $F\hat{O}V$ and the value of w set to be equal to $b_1 + b_2$ in $F\hat{O}V$. Let w' and h' denote the values for w and h respectively in a unit density deployment. Under this deployment density, the initial state for the camera system will be such that the coverage resolution offered by any camera cannot be increased any further. As a result, whenever a camera has been assigned for tracking, the system cannot be completely covered any more. With this deployment density as reference, we define the camera network density to be D when $w = \frac{w'}{\sqrt{D}}$ and $h = \frac{h'}{\sqrt{D}}$.

The cameras are deployed at a height 8 feet from the ground. Under the highest coverage resolution offered by each camera, the values of b_1, b_2 and l for the camera's FOV (Fig. 2) are approximately 25 feet, 3 feet and 29 feet respectively. Under the lowest coverage resolution offered by each camera, the values of b_1, b_2 and l are equal to 1 foot, 1 foot and 2 feet respectively. As a result in a unit density deployment, w and d are set to be equal to 25 feet and 29 feet respectively. We consider a single target in the system at a time that is introduced at different locations, mainly near the center of the network. We consider two different target speeds, 2.5mph (A walking target) and 7.5mph (A running target).

B. Impact of A_L on coverage resolution for $D > 1$

Here, we analyze the impact of the size of the area considered for reconfiguration (A_L) on the coverage resolution

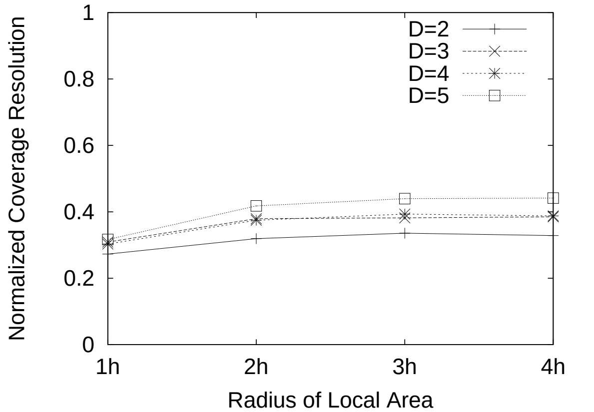

Fig. 5. Impact of A_L on coverage resolution. The y-axis shows the normalized coverage resolution for all points in A_M.

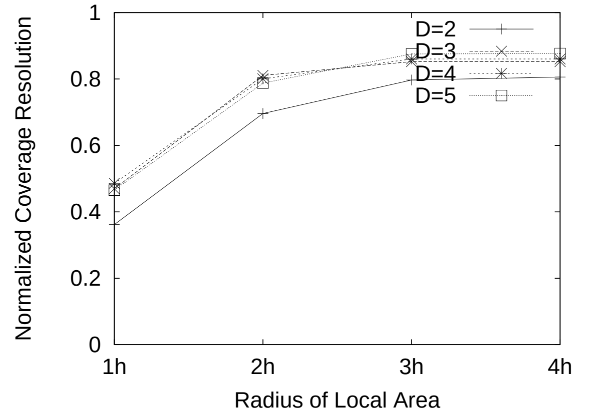

Fig. 6. The impact of A_L on coverage resolution. The y-axis shows the 75^{th} percentile of the normalized coverage resolution among all points in A_L.

at various network densities greater than 1. Recall that at camera density of 1, the network is observing the entire area at minimum coverage resolution and as a result the area cannot be completely covered when one or more cameras are *lost* for tracking a target. Hence for the case of unit network density, we subsequently evaluate the loss of coverage as a function of A_L.

In Fig. 5, we show the impact of A_L on coverage resolution. The area A_L is considered as a circle of radius zh around the location of the camera assigned to track the target, where h is the separation between adjacent rows of cameras. The x-axis in Fig. 5 represents the radius of this circle for different values of z. The coverage resolution is normalized with respect to the coverage resolution of A_M in the initial state ($R_I(A_M)$) as a reference point. Let $R(A_M)$ denote the coverage resolution achieved in simulation for a particular trial of the experiment. The normalized coverage resolution for that trial is then equal to $\frac{R(A_M)}{R_I(A_M)}$. The values for normalized coverage resolution depicted in Fig. 5 are based on an average over 1000 trials for each value of z and D. We observe that the improvement in resolution coverage starts decreasing significantly at values of $z > 2$ and this remains true at all the evaluated network densities.

Note that by definition $R(A)$ reflects the lowest coverage resolution among all points in an area A. In order to understand the coverage resolution for a majority of points within A_L, in Fig. 6 we use the 75^{th} percentile of coverage resolution among all points in A_L as a metric for evaluating the impact of locality size on coverage resolution. In other words only 25% of points within A_L have a lower coverage resolution than those shown in Fig. 6. We observe from Fig. 6 that coverage resolution for a majority of points is significantly higher than those shown in Fig. 5.

C. Impact of A_L on loss of coverage at $D = 1$

In this subsection, we evaluate the impact of the size of the area considered for reconfiguration (A_L) on the loss of coverage at $D = 1$. Note that at $D = 1$, the cameras are unable to zoom out or tilt up to cover any more space. The

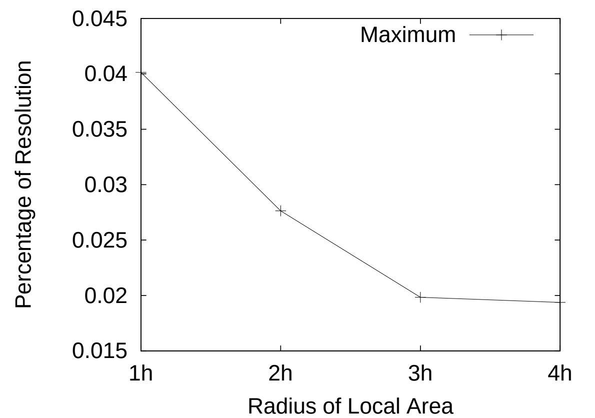

Fig. 7. The size of the maximum gap given as a percentage of the area of the original gap created by the *loss* of the tracking camera in a network with unit density.

cameras can only pan towards a gap to decrease the size of the largest gap and spread the uncovered space more evenly across the network. With respect to the steps $R1 - R4$ for the reconfiguration algorithm, we note that steps $R2$ and $R3$ are inactivated. Also step $R4$ cannot result in any decrease in unobserved area. So we use a heuristic and modify $R4$ so that the pan parameter of a camera is adjusted to move towards the largest adjacent gap only if the area of that gap can decrease by 1 square foot while not increasing the total uncovered area. The algorithm terminates when no camera's pan parameters can be adjusted in a single round or in a maximum of 10 rounds to avoid oscillations.

Fig. 7 shows the impact of A_L on the size of the largest gap in the area being monitored. We notice that as the size of the local area increases, the size of the largest gap also decreases. This is due to the increasing number of smaller gaps throughout the larger area.

D. Impact of target speed on stability of the system

The speed of a moving target is expected to influence the frequency of reconfiguration events. We simulate targets at two

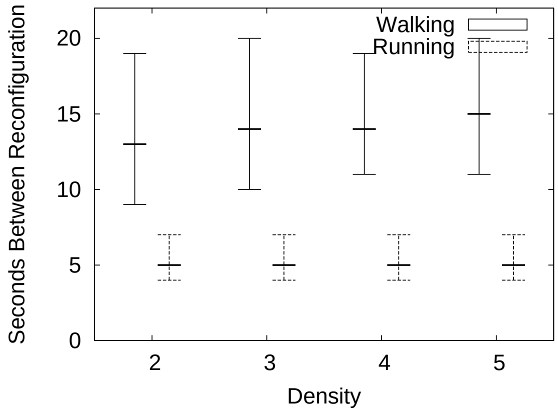

Fig. 8. Time between reconfiguration events for walking and running targets

speeds moving through the network and determine the time between successive reconfiguration events. Fig. 8 shows the 25^{th}, 50^{th} and 75^{th} percentile of time between reconfiguration events for 2 target speeds, a walking speed and a running speed, at different camera densities. For example at $D = 2$, there is an average of 5 seconds between two reconfiguration events at a running speed which implies that the time for reconfiguration certainly needs to be smaller than this and greater the difference, better the stability of the system. This signifies the importance of having smaller reconfiguration times and highlights the merit of choosing a local approach and performing just a single PTZ adjustment per reconfiguration event.

V. Conclusions and Future Work

We have considered a network of PTZ cameras used for monitoring a large region and described a local algorithm that can track targets at a high resolution while maximizing the coverage resolution at all non-target points. We evaluated the performance of our algorithm in simulations and analyzed the impact of network density and the size of the area over which the reconfiguration is performed on the achievable coverage resolution. We observe that by progressively increasing the size of the locality over which the reconfiguration is performed, the improvement in coverage resolution starts decreasing significantly, thus highlighting the merits of a localized approach. As a result, we are able to track targets at a high resolution, observe the entire space at an acceptable resolution, and keep the observing cameras stable for long periods of time by only performing the reconfiguration in a local area around the target.

Completely decentralized systems for determining coverage where each camera successively makes adjustments and communicates this change to neighboring cameras before converging to an optimum configuration [7] have an advantage in that they do not rely on a single camera for computing the reconfiguration parameters. However, when dealing with mechanical systems like PTZ motors that consume significant time to affect a change, it is more suitable to adopt a strategy where the nodes are able to reach a final configuration within a single round. Hence we have opted for a non-iterative strategy that relies on computation at a single node and then propagating the new parameters back to the respective cameras.

Our initial analysis has thrown open several interesting avenues for further research. In this paper, we have considered the case of a single target being tracked. We would like to modify our algorithm to handle multiple targets in the system and evaluate its performance. In the future, we would also like to demonstrate a working prototype of our dynamic reconfiguration system at a small scale, using a wireless network of PTZ cameras integrated with local processing capability. We are also interested in integrating an active camera based face recognition system such as [9], [10] with our reconfiguration system for maintaining coverage at high quality while performing human identification.

In this paper, we have assumed the existence of a target detection algorithm that is also able to localize the target. We also rely on a network level service for reliably electing a camera that is responsible for tracking a target. Designing network services for target detection and camera assignment in a way that is appropriate for a distributed implementation, and integrating those with the reconfiguration algorithm presented in this paper is also a subject of our ongoing research.

References

[1] Q. Cai and J. Aggarwal, “Tracking human motion in structured environments using a distributed-camera system,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 11, pp. 1241 –1247, Nov. 1999.

[2] R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson, “Decentralized cooperative aerial surveillance using fixed-wing miniature uavs,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306 –1324, 2006.

[3] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, “Algorithms for cooperative multisensor surveillance,” *Proceedings of the IEEE*, vol. 89, no. 1, pp. 1456 – 1477, October 2001.

[4] Y. Yao, C. Chen, B. Abidi, D. Page, A. Koschan, and M. Abidi, “Sensor planning for automated and persistent object tracking with multiple cameras,” in *International conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[5] 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,” in *International conference on Advanced Video and Signal Based Surveillance*, 2008.

[6] H. Jin and G. Qian, “Robust Multi-Camera 3D People Tracking with Partial Occlusion Handling,” in *International conference on Acoustics, Speech and Signal Processing*, 2007.

[7] M. Huffman and M. Wittke and J. Hahner and C. Muller-Schloer, “Spatial Partitioning in Self-Organizing Smart Camera Systems,” *IEEE Journal on Selected Topics in Signal processing*, vol. 2, no. 4, pp. 480–492, 2008.

[8] M. Schwager, B. Julian, and D. Rus, “Optimal coverage for multiple hovering robots with downward-facing cameras,” in *Proceedings of the International Conference on Robotics and Automation (ICRA 09)*, May 12–17 2009, pp. 3515–3522.

[9] F. W. Wheeler and R. L. Weiss and P. H. Tu, “Face recognition at a distance system for surveillance applications,” in *Fourth IEEE International Conference on Biometrics: Theory Applications and Systems*, 2010, pp. 1–8.

[10] N. Krahnstoever, S.-N. L. T. Yu, K. Patwardhan, and P. Tu, “Collaborative real-time control of active cameras in large scale surveillance systems,” in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.