ENSURING NETWORK CONNECTIVITY OF UAV'S PERFORMING VIDEO RECONNAISSANCE

Nicholas R. Gans[a], John M. Shea[b], P. Barooah[a] and Warren. E. Dixon[a]

***Abstract*—Current and emerging military missions require image-feedback from camera-equipped assets. Motivated by mission scenarios and sensor restrictions, operations may require the collaboration of assets over an ad-hoc network. The development in this paper represents the first efforts to examine the problem of balancing trade-offs between asset/sensor cone positioning to satisfy mission requirements and network requirements such as maintaining network connectivity. To address the trade-offs between asset positioning and network connectivity, a prioritized task-function based guidance law is developed for a simple scenario containing three assets. One developed task-function maintains a communication network by ensuring the distance between the UAV's does not exceed a critical threshold. Additional task-functions enable assets to keep targets of interest in the image cone by regulating image features derived from the camera view. Simulation results are provided to examine the behavior of the assets for different configurations of objects observed by the asset cameras.**

I. INTRODUCTION

Military assets are equipped with imaging sensors in many scenarios, including image-based guidance, navigation, and control, position and orientation (i.e., pose) estimation of assets for red and blue force tracking, environment mapping, etc. A historical problem in image-based estimation and control literature is controlling the motion of the camera to ensure that targets of interest stay in the field-of-view (FOV) (cf. [1]–[4]). This problem is especially difficult when the camera is monitoring an adversarial target in a dynamic environment, or when it is mounted on a vehicle with motion constraints. A practical approach to alleviate the FOV problem is to use multiple (non-stereo vision) imaging sources [5], [6], such as cooperative camera-equipped assets. A novel method to control the motion of a single camera to keep multiple assets in the FOV was presented in [7].

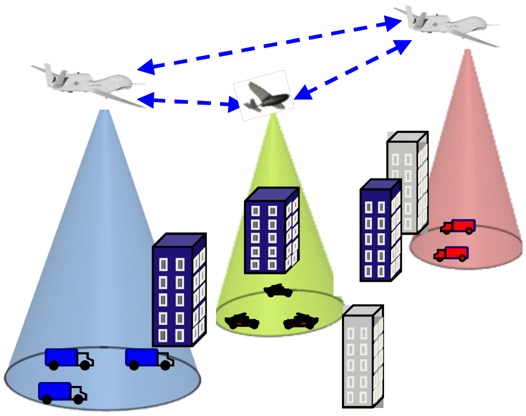

Fig. 1. Networked collection of UAV's performing image-based red force and blue force tracking in an urban environment.

The use of cooperative assets is a well accepted approach in military missions. Current and emerging military missions (e.g., wide area surveillance, bomb damage assessment, contaminant and dispersion monitoring, search and rescue/destroy) can exploit coordination between multiple assets for improved efficiency. In many applications, such as tactical operations, portions of the network linking the collaborative assets may entail a mobile ad hoc network (MANET). Assets operating within a MANET are faced with challenges, such as maintaining connectivity due to the changing topology and link conditions of the network. These problems are exacerbated by high mobility, which is characteristic of airborne networks.

Fig. 1 illustrates an example scenario of a network of imaging assets. In this scenario, a network of unmanned air vehicles (UAV's) are tracking red forces and blue forces in an urban scenario. The imaging goals are challenging because the urban environment limits the FOV by an asset, and the cameras are attached to UAV's with motion constraints. Coordinating the assets is challenging because network connectivity must be maintained in the presence of path loss, shadowing,

N. Gans is with the National Research Council and Air Force Research Lab, Eglin Air Force Base, FL, ngans@ufl.edu; J. Sheah is with the Dept. of Electrical & Computer Engineering, Univ. of Florida, Gainesville, FL, jshea@ufl.edu; P. Barooah and Warren Dixon are with the Dept. of Mechanical and Aerospace Engineering, Univ. of Florida, Gainesville, FL, {pbarooah, wdixon}@ufl.edu

[1] This research is supported in part by AFOSR contract numbers F49620-03-1-0381, F49620-03-1-0236, and FA9550-07-10456, AFRL contract number FA4819-05-D-0011,, and by the US Department of Energy grant number DE-FG04-86NE37967. This work was accomplished as part of the DOE University Research Program in Robotics (URPR), and was performed, in part, while Nicholas Gans held a National Research Council Research Associateship Award at the Air Force Research Laboratory.

978-1-4244-2677-5/08/$25.00 ©2008 IEEE

and/or multi-path fading. Even with perfect location information, maintaining a fully connected network topology can be challenging. The potential tactical advantages of the scenario in Fig. 1 motivate the need for directed methods that manage trade-offs between asset/sensor cone positioning to satisfy mission requirements and network requirements to ensure effective collaboration between the assets. Yet, literature that focuses on such issues appears sparse.

This paper presents a first attempt at addressing some of the challenges imposed by balancing asset positioning for image tasks against asset positioning for efficient network operation. Motivated by the scenario depicted in Fig. 1, the imaging task involves a collection of assets maintaining sets of moving objects in the camera FOV's, while also positioning the assets to ensure network connectivity. For example, one asset observes the blue force, while the other assets monitor sub-groups of the red force. For this initial result, the development neglects issues such as the dynamics of the asset (i.e., the asset is assumed to have no motion constraints) and the network is restricted to three nodes. The network is modeled as undirected, and successful communication between two assets is assumed possible if two assets are within some specified maximum-link distance.

To achieve imaging versus network trade-offs, a series of low dimensional task-functions are defined, based on current image features and the distance to the nearest neighbor in the network. For the image-based task functions, the desired task-function velocity is mapped to a time varying feature velocity through corresponding Jacobians. These task-function Jacobians are underdetermined and are suitable for task-priority kinematic control [8], [9]. The time varying image feature velocity is mapped to camera motions through image-based visual servoing (IBVS) methods [10]. The resulting controller allows features to move within the image, and the camera will move to keep the features in the FOV. To maintain network continuity, we propose an additional task of regulating the position of each camera to maintain distance to its nearest neighbor. If each camera is within a maximum distance to its nearest neighbor, the network of three nodes will stay simply connected.

The problem of group coordination of mobile robots for connectivity maintenance has been studied in several papers (cf. [11]–[14]). Most of these studies are concerned with the robots moving toward a goal destination while maintaining wireless connectivity. In other words, the position of the mobile agents – if there were no need to maintain connectivity – would tend to certain fixed location in space depending on the motion objective. In contrast, we study the situation when there are no limiting goal positions. Rather, the unpredictable motion of the targets may cause the UAV's to move in a way that will sever connectivity in the absence of a connectivity maintaining control law.

Maintaining connectivity among moving agents with second order dynamics has been examined in [15], [16]. However, the uncontrolled motion of the agents in [15] was only due to initial velocities and second-order dynamics, not due to the need for maintaining targets within the FOV. A similar problem set up was also considered in [16] for first order dynamic agents. In contrast to these studies, we examine the situation when sensing and maintaining connectivity are in conflict, e.g., when motion of the targets may cause the UAV's to move away from one another while maintaining connectivity may require them to stay close. However, we ignore the dynamics of the agents for the present study.

II. CAMERA AND NETWORK MODEL

Consider a camera with coordinate frame $\mathcal{F}_c(t)$. The camera views a collection of k feature points in front of the camera. These points have coordinates $M_i(t) \in \mathbb{R}^3$ defined as

$$M_i = [X_i, Y_i, Z_i]^T, \ \forall i \in \{1 \ldots k\}$$

in the camera frame. An image of the points is captured, resulting in a projection to a set of points in the image plane. These image points are given by the coordinates $m_i(t) \in \mathbb{R}^2$

$$\breve{m}_i = [\frac{X_i}{Z_i}, \frac{Y_i}{Z_i}]^T = [x_i, y_i]^T, \ \forall i \in \{1 \ldots k\}$$

with velocity $\dot{m}_i(t) \in \mathbb{R}^2$ in the image plane given by

$$\dot{m}_i = [\dot{x}_i, \dot{y}_i]^T., \ \forall i \in \{1 \ldots k\}$$

Given the collection of k feature points, along with their coordinates and velocity vectors, a state position vector $m(t)$ and state velocity vector $\dot{m}(t)$ are defined as

$$\begin{aligned} m &= [m_1^T, m_2^T, \cdots, m_k^T]^T \\ \dot{m} &= [\dot{m}_i^T, \dot{m}_2^T, \cdots, \dot{m}_k^T]^T. \end{aligned}$$

The features points velocity is given as a function of the camera velocity $v_c(t) = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z,]^T \in \mathbb{R}^6$ by the relationship

$$\dot{m} = Lv_c,$$

where $L(t) \in \mathbb{R}^{2k \times 6}$ is the image Jacobian, $v_x(t)$, $v_y(t)$ and $v_z(t)$ describe the linear velocity of the camera, and $\omega_x(t)$, $\omega_y(t)$ and $\omega_z(t)$ are the angular velocity of the camera. The image Jacobian for the feature points, $L(t) \in \mathbb{R}^{2k \times 6}$, is given by concatenating a set of k

978-1-4244-2677-5/08/$25.00 ©2008 IEEE

submatrices $L_i(t) \in \mathbb{R}^{2\times 6}$ [17], with $L_i(t)$ given as

$$L_i = \left[\begin{array}{cccccc} \frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & -x_i y_i & 1 + x_i & -y_i \\ 0 & \frac{1}{Z_i} & \frac{y_i}{Z_i} & -1 - y_i^2 & x_i y_i & x_i \end{array} \right].$$

In the case that the feature points are not static in an inertial frame world frame (e.g., the feature points are tracked on moving objects), the time derivative of the feature points is given by

$$\dot{m} = L v_c + \varepsilon, \tag{1}$$

where $\varepsilon(t)$ is an unknown, bounded function.

The network of N assets is modeled as a graph $G = (V, E)$ with N nodes. Each asset j, $j \in [1, \ldots, N]$ is a node of the graph, and is located at a position T_j. There is an edge (i, j) between two assets i and j, $i \neq j$, if the distance between them is less than a prespecified positive number r, i.e., if $d_{ij} < r$, where the distance is defined as

$$d_{ij} = \|T_i - T_j\|,$$

and r is a communication range. This paper considers only the case $N = 3$. In this case, the graph remains connected if each node has an adjacency of at least one, i.e. each asset is within a radius r to at least one other asset.

III. TASK FUNCTION-BASED KINEMATIC CONTROL

Let $\phi(t) \in \mathbb{R}^n$ denote some task function of the feature point coordinates $m_i(t)$ as

$$\phi = f(m_1, \ldots, m_k)$$

with derivative

$$\dot{\phi} = \sum_{i=1}^{k} \frac{\partial f}{\partial m_i} \dot{m}_i = J(m)\dot{m},$$

where $J(m) \in \mathbb{R}^{n\times 2k}$ is the task Jacobian matrix. The task functions in this paper are of dimension $n \leq 2$.

The task is to drive the feature points along a desired velocity $\dot{m}_\phi(t)$ such that $\phi(t)$ follows a desired trajectory $\phi_d(t)$. Given the underdetermined structure of the Jacobian matrix, there are infinite solutions to this problem. The typical solution based on the minimum-norm approach [18] is given as

$$\begin{aligned} \dot{m}_\phi &= J^\dagger \left[\dot{\phi}_d - \lambda(\phi - \phi_d) \right] \\ &= J^T \left(J J^T \right)^{-1} \left[\dot{\phi}_d - \lambda(\phi - \phi_d) \right], \end{aligned} \tag{2}$$

where λ is a positive scalar gain constant, and $J^\dagger(m) \in R^{2k\times n}$ denotes the minimum-norm general inverse of $J(m)$. Based on (2), the camera velocity $v_c(t)$ is designed as

$$v_c = L^+ \dot{m}_\phi = \left(L^T L \right)^{-1} L^T \dot{m}_\phi,$$

where $L^+(t) \in \mathbb{R}^{6\times 2k}$ denotes the least squares general inverse used for $L(t)$. Since $J(m)$ is underdetermined (i.e. more columns than rows), and $L(t)$ is overdetermined (i.e. more rows than columns), the general inverses have different solutions and care is taken to denote them differently.

Multiple task functions can be combined in various ways. Since the task function Jacobians are undetermined, one method is to choose one task as primary, and project the other tasks into the null space of the primary task derivative [8], [18] as

$$\begin{aligned} v_c &= L^+ \left(\dot{m}_a + (I - J_a^\dagger J_a)\dot{m}_b \right) \\ &= L^+ \left(J_a^\dagger \dot{\phi}_a + (I - J_a^\dagger J_a) J_b^\dagger \dot{\phi}_b \right), \end{aligned} \tag{3}$$

where $J_a(m_a)$ and $J_b(m_b)$ are the task Jacobian matrices with respect to $\phi_a(t)$ and $\phi_b(t)$, respectively.

The approach in (3) will prevent the systems from competing and negating each other, as the primary task will always be accomplished. Lower priority control tasks will be achieved to the extent that they do not interfere with higher priority tasks. Tertiary, quaternary, etc. tasks can be prioritized by repeating this process and projecting each subsequent task into the null space of the preceding task Jacobians.

IV. CONTROL DEVELOPMENT

In this section, four task functions are presented as part of a distributed, task-priority kinematic controller. The objective is to keep three sets of feature points within the FOV of three cameras, while maintaining a communication network between the camera equipped assets. Each camera is dedicated to observing a single set of feature points.

The first task function regulates the distance to the nearest camera, which will maintain a network connection for three cameras. Two task functions are designed to regulate the mean and variance of feature point coordinates. Regulating the mean at the camera center will keep the feature points centered in the FOV. Regulating the variance will restrict the distance between the feature points and keep features away from the edge of the FOV. The third task function maximizes motion perceptibility, which keeps the image Jacobian well conditioned and ensures desired image velocities can be met. These task functions are cascaded through null space projection and mapped to camera velocity, as described Section III.

Chebyshev's inequality proves that at least 75% of all values are within two standard deviations of the mean, and at least 89% of values are within three standard deviations. For a normally distributed random process, these limits are tighter, such that approximately 95%

of all values will be with two standard deviations, and 99.7% of all values will be within three standard deviations. Consider a camera with a 512x512 pixel FOV and normally distributed feature points in the image plane. Regulating the mean of the feature point coordinates to the image center and the variance of the feature point coordinates to 128^2 will ensure that at least 95% of all points are in the FOV. For arbitrary distribution of feature points (e.g. uniformly distributed) points, regulating the variance to 86^2 will ensure that at least 89% of all points are in the FOV.

A. TASK FUNCTION TO MAINTAIN NETWORK CONNECTIVITY

The primary concern of this paper is to maintain network communication between multiple assets as they perform individual tracking tasks. For three camera-equipped assets modeled as a proximity graph, maintaining network connectivity can be modeled as regulating the distance from each asset to its nearest neighbor. The network will be remain simple connected if every asset remains within a bounded distance r to at least one other asset. To give the assets freedom of motion, it is desirable that the distance regulation function not be active until the distance d_{ij} is beyond a certain threshold $\underline{r} < r$. To this end, we propose the use of a smooth task function inspired by the p-times smooth bump function given in [19]. This smooth task function is given by

$$\phi_d = \sum_{i=1}^{3} \begin{cases} \frac{1}{p!}\int_{\underline{r}}^{d_i} (\tau - \underline{r})^p d\tau & \text{if } \tau > \underline{r} \\ \frac{1}{p!}\int_{-\underline{r}}^{d_i} (-\underline{r} - \tau)^p d\tau & \text{if } \tau < -\underline{r} \\ 0 & \text{else} \end{cases}$$

where $d = [d_1,\ d_2,\ d_3]^T$ is the coordinates of the nearest neighbor in the camera frame, $-\underline{r},\ \underline{r} \in \mathbb{R},\ -\underline{r} < \underline{r}$ define a window wherein $\phi_d = 0$, and $p \in \mathbb{R}$ gives the smoothness of the function ϕ_d. Note that $\phi_d(t) = 0$ if $d_i \in [-\underline{r}, \underline{r}]\ \forall i$. Furthermore, $\phi_d(t)$ can be solved in closed form, evaluates to a positive definite function that can be differentiated p-times, and $\frac{\partial^p \phi_d}{\partial x^p} = \pm 1$ for $\phi_d \notin [-\underline{r}, \underline{r}]$. For example, for $p = 2$

$$\frac{1}{2}\int_{\underline{r}}^{d_i} (t - \underline{r})^2 dt = \frac{d_i^3}{6} - \frac{\underline{r}}{2} d_i^2 + \frac{\underline{r}^2}{2} d_i - \frac{\underline{r}^3}{6}$$
$$= \frac{1}{6}(d_i - \underline{r})^3.$$

These characteristics make the function useful for systems with high order dynamics. For the kinematic systems in this paper, $p = 1$ is sufficient, in which case the polynomial evaluates to

$$\int_{\underline{r}}^{d_i} (t - \underline{r}) dt = \frac{1}{2}(d_i - \underline{r})^2$$

which is the familiar quadratic. The derivative of $\phi_d(t)$ can be expressed as

$$\begin{aligned} \dot{\phi}_d &= \sum_{i=1}^{3} \frac{\partial \phi_d}{\partial d_i} \dot{d}_i = J_d \dot{d} \\ &= J_d (v_c + \varepsilon_d), \end{aligned} \quad (4)$$

where $J_d(t) \in \mathbb{R}^{1 \times 6}$ is a task function Jacobian, and $\varepsilon_d \in \mathbb{R}^6$ is a disturbance due to the velocity of the nearest neighbor. The Jacobian $J_d(t)$ is defined as

$$J_d = [\mathrm{sgn}(d)^T, 0_3],$$

where $\mathrm{sgn}(\cdot)$ is the vector signum function, and $0_3 \in \mathbb{R}^{1 \times 3}$ is a zero row-vector.

The objective is to regulate $\phi_d(t) \to 0$, under the assumption that the current velocity of an asset's nearest neighbor is available through the connected communication network (i.e., ε_d is known). Based on the task function time derivative in (4), a stabilizing camera velocity can be designed as

$$v_{cd} = -\lambda_d J_d^{\dagger} \phi_d - \varepsilon_d, \quad (5)$$

where λ_d is a positive scalar gain constant. Combining (4) and (5) yields the exponentially stable system

$$\dot{\phi}_d = -\lambda_d \phi_d.$$

B. TASK FUNCTIONS T0 MAINTAIN TARGETS IN THE FOV

A novel method was presented in [7] to control a camera such that multiple moving targets could be kept in the FOV. This method was based on kinematic control using multiple, prioritized task functions, as discussed in Section III. The task functions will be briefly introduced here, and the reader is referred to [7] for further details.

Define a task function $\phi_m(t) \in \mathbb{R}^2$ as the sample mean

$$\phi_m = \frac{1}{k} \sum_{i=1}^{k} m_i = \bar{m}$$

with derivative

$$\dot{\phi}_m = \frac{1}{k} \sum_{i=1}^{k} \frac{\partial \phi_m}{\partial m_i} \dot{m}_i = J_m \dot{m}$$

where $J_m(t) \in R^{2 \times 2k}$ is the task function Jacobian. A stable PID controller can be developed to regulate $\phi_m(t)$ to a constant set point ϕ_{md}

$$\dot{m}_m = -J_m^{\dagger} \left(\lambda_m \phi_{me} + \lambda_{mi} \int_0^t \phi_{me} dt + \lambda_{md} \frac{d}{dt} \phi_{me} \right), \quad (6)$$

978-1-4244-2677-5/08/$25.00 ©2008 IEEE

where $\phi_{me}(t) = \phi_m(t) - \phi_{md}$, and $\lambda_{mi}, \lambda_{md} \in R^+$ are constant gains.

Define a task function $\phi_v(t) \in \mathbb{R}^2$ as the sample variance

$$\phi_v = \frac{1}{k}\sum_{i=1}^{k}\left[\begin{array}{c}(x_i - \bar{x})^2\\(y_i - \bar{y})^2\end{array}\right],$$

where $\bar{x}(t)$ and $\bar{y}(t)$ are the mean of all the x and y components of $m_i(t), i \in \{1 \ldots k\}$, respectively. The derivative of $\phi_v(t)$ is given by

$$\dot{\phi}_v = J_v \dot{m}$$

where $J_v(t) \in \mathbb{R}^{2\times 2k}$ is a task function Jacobian. Regulation of the variance to a set point ϕ_{vd} can be accomplished by a PID feedback control of the form

$$\dot{m}_v = -J_v^\dagger\left(\lambda_v \phi_{ve} + \lambda_{vi}\int_0^t \phi_{ve}dt + \lambda_{vd}\frac{d}{dt}\phi_{ve}\right) \quad (7)$$

where $\phi_{ve}(t) = \phi_v(t) - \phi_{vd}$, and $\lambda_{vi}, \lambda_{vd} \in \mathbb{R}^+$ are constant gains.

Motion perceptibility is a scalar value first presented in [20]. A high value of perceptibility ensures the image Jacobian is well conditioned, which is essential for regulating the vision-based task functions. As described in [7], the perceptibility task function $\phi_p(t) \in \mathbb{R}^2$ is defined as

$$\phi_p = \frac{1}{\sum\limits_{i=1}^{k} x_i^2 + y_i^2 + 2x_i^2 y_i^2 + \left(y_i^2 + 1\right)^2 + \left(x_i^2 + 1\right)^2}.$$

Regulating $\phi_p(t) \to 0$ will increase perceptibility. The derivative of $\phi_p(t)$ can be expressed as

$$\dot{\phi}_p = J_p(m)\dot{m}$$

where $L(t)$, $v_c(t)$ and $\varepsilon(t)$ are introduced in (1), and $J_p(m) \in \mathbb{R}^{1\times 2k}$ is the task function Jacobian for perceptibility.

To regulate $\phi_p(t) \to 0$, a proportional control law is defined as

$$\dot{m}_p = -\lambda_p J_p^\dagger \phi_p, \quad (8)$$

where λ_p is a positive scalar gain constant.

C. CASCADED CAMERA CONTROL LAW

The control objective of this paper is to design a camera controller that maintains a set of feature points within the FOV's of several independent assets. The controller is decentralized, and each asset operates under an identical control law. In addition to maintaining the view of the targets, each camera must remain within a maximum distance from its nearest neighbor in order to maintain a communication network.

Maintaining the connectivity of the network is the primary task. Regulating the mean to the image center is chosen as the secondary task in order to keep the feature points centered in the FOV. Regulating the variance to a constant is chosen as the tertiary task to restrict the distance between the feature points and the image center. These two tasks ensure features remain in the FOV. High perceptibility will allow these two tasks to work more efficiently by ensuring larger available feature velocities at lower camera velocities. For this reason, increasing perceptibility is chosen as the quaternary task.

The designed feature velocities given in (5)-(8) are used in the null-space projection camera velocity (3) to give the overall controller for each camera as

$$\begin{aligned} v_c = v_{cd} + \left(I - J_d^\dagger J_d\right) L^+ \cdot \\ \left(\dot{m}_m + \left(I - J_m^\dagger J_m\right)\left[\dot{m}_v + \left(I - J_v^\dagger J_v\right)\dot{m}_p\right]\right). \end{aligned}$$

V. SIMULATION RESULTS

In these simulations, three cameras observe six rigid, square objects. Each camera is responsible for observing two of the square objects, and the corners of the squares give eight feature points for each camera to track. Each pair of objects move with a sinusoidal linear and angular velocity, independent of the other pairs. This simulation mimics the case of airborne cameras tracking sets of ground vehicles. Each camera has a resolution of 512×512. For each camera controller,the mean was regulated to the image center, $[256, 256]^T$. The variance of the points was regulated to $[100^2,\ 100^2]^T$, i.e. a standard deviation of 100 pixels. The simulation was executed for 10 seconds at 30 frames per second.

The first simulation does not include any attempt to regulate the distance between the cameras, i.e. the cameras are free to move as necessary to track the targets. Fig. 2 shows the final 3D positions of the three targets and the three pairs of targets. Camera 2 ends up far from the other cameras. Fig. 3 shows the views of the three cameras, including the final image of the tracked targets and the trajectory the corner points traced in the image over time. From the trajectory curves, it can be seen that the objects remain in view of each camera. The dashed ellipse and square represent the final values of the variance and mean, while the solid ellipse and star represent the goal variance and mean, though they overlap closely in these figures.

The second simulation includes the regulation of distance between the cameras. Fig. 4 shows the final 3D positions of the three targets and the three pairs of targets. While the targets all end at the same locations, the cameras end up much closer to each other. Fig. 5

978-1-4244-2677-5/08/$25.00 ©2008 IEEE

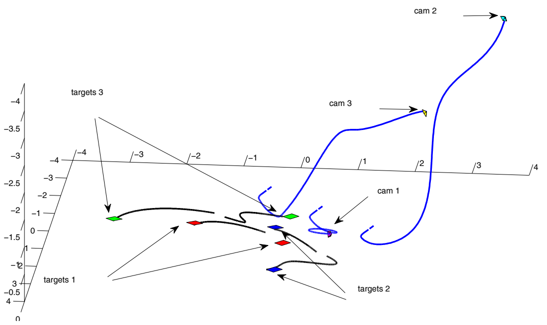

Fig. 2. Final 3D locations of the three cameras and the three sets of targets when inter-camera distance is not regulated.

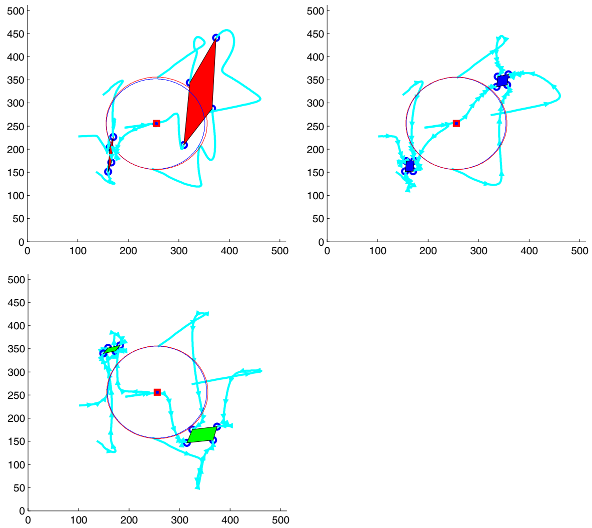

Fig. 3. The three camera views and trajectories over time when inter-camera distance is not regulated.

shows the views of the three cameras. From the trajectory curves, it can be seen that regulating the distance between cameras makes the tracking task harder, and one points is temporarily lost from view in camera one. However, tracking is successful in that no target ever fully leaves the field of view.

Fig. 6 shows values of the task functions over time for the case when inter-camera distance is not regulated, and Fig. 7 when distance is regulated. As expected, when distance is regulated, the mean and variance tracking error is larger, but the final distance between the cameras is much smaller.

VI. Conclusions and Future Work

This paper presents the results of an initial investigation into balancing two competing surveillance tasks,

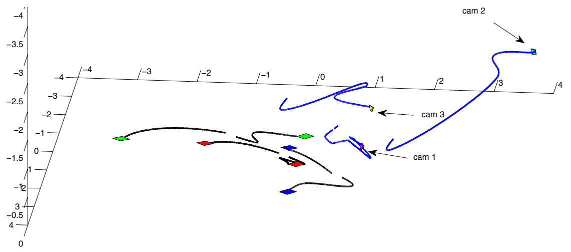

Fig. 4. Final 3D locations of the three cameras and the three sets of targets when inter-camera distance is not regulated.

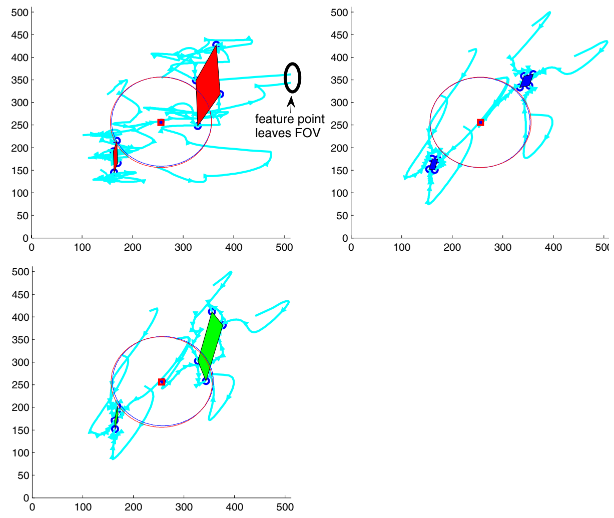

Fig. 5. The three camera views and trajectories over time when inter-camera distance is regulated.

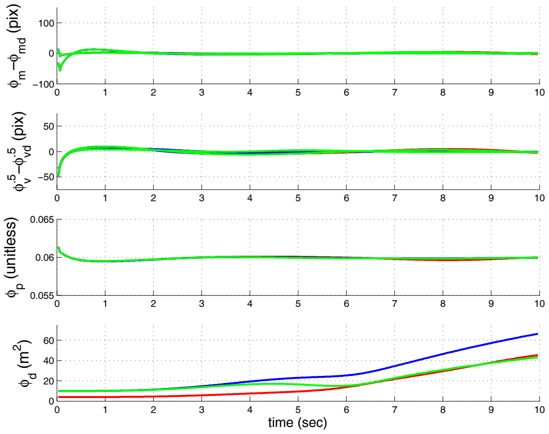

Fig. 6. Values of task functions over time when distance between cameras is not regulated

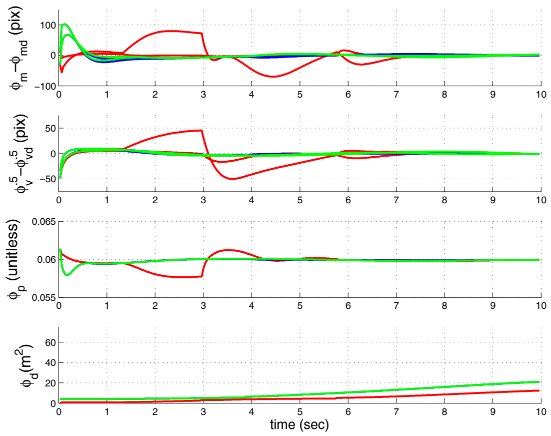

Fig. 7. Values of task functions over time when distance between cameras is regulated

using multiple assets to monitor multiple, moving targets while maintaining a network communications between the assets. This problem is visualized as multiple UAV's equipped with cameras tracking several groups of moving vehicles, with a network modeled as a proximity graph.

To achieve these tasks, a series of prioritized, underdetermined task functions were developed. Maintaining network connectivity can be ensured by regulating the distance of each camera to its nearest neighbor. Targets can be kept in field of view through regulating mean and variance of the targets' features in the image. A third task function seeks to maximize motion perceptibility. There is no specific goal image or goal pose for the cameras, rather the underdetermined nature of the task functions allows the camera to move as necessary to regulate the task functions and keep objects in the FOV while maintaining network connectivity.

There are several avenues of future work. No formal stability analysis has been presented of the closed loop system. Simulations are very promising, but a proper analysis can determine whether, or under what conditions, stability can be achieved for all tasks. There are also numerous other task functions that could be used. For instance, it may be desirable to maintain a certain distance or orientation with respect to the tracked objects. The network was limited to three nodes, which allowed for the use of a simple distance regulation to maintain the network. This methodology must be extended to a larger, possibly unlimited, number of assets. This will require a better metric for measuring and regulating network connectivity.

References

[1] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. Robot. Automat.*, vol. 17, no. 4, pp. 507–515, 2001.

[2] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. Robot. Automat.*, vol. 18, no. 4, pp. 534–549, 2002.

[3] N. Gans and S. Hutchinson, "Stable visual servoing through hybrid switched-system control," *IEEE Trans. Robotics*, vol. 23, no. 3, pp. 530–540, 2007.

[4] J. Gaspar, N. Winters, and J. Santos-Victor, "Vision-based navigation and environmental representations with an omni-directional camera," *IEEE Trans. Robotics*, vol. 16, no. 6, pp. 890–898, 2000.

[5] G. Flandin, F. Chaumette, and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing," in *Proc. IEEE International Conference on Robotics and Automation ICRA '00*, vol. 3, pp. 2741–2746 vol.3, 2000.

[6] W. Dixon, E. Zergeroglu, Y. Fang, and D. Dawson, "Object tracking by a robot manipulator: a robust cooperative visual servoing approach," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 1, pp. 211–216 vol.1, 2002.

[7] N. R. Gans, G. Hu, and W. E. Dixon, "Keeping objects in the field of view: An underdetermined task function approach to visual servoing," in *Proc. IEEE Multi-Conf. Systems and Control*, pp. 432–437, 2008.

[8] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robotics Research*, vol. 9, pp. 3–15, 1987.

[9] G. Antonelli and S. Chiaverini, "Kinematic control of platoons of autonomous vehicles," vol. 22, pp. 1285–1292, Dec 2006.

[10] F. Chaumette and S. Hutchinson, "Visual servo control part I: Basic approaches," *IEEE Robotics and Automation Mag.*, vol. 13, no. 4, pp. 82–90, 2006.

[11] A. R. Wagner and R. C. Arkin, "Communication-sensitive multi-robot reconnaissance," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2480–2487, 2004.

[12] J. Sweeney, T. J. Brunette, Y. Yang, and R. A. Grupen, "Coordinated teams of reactive mobile platforms," in *International Conference on Robotics and Automation (ICRA)*, pp. 99–304, 2002.

[13] G. A. S. Pereira, A. K. Das, V. Kumar, and M. F. M. Campos, "Decentralized motion planning for multiple robots subject to sensing and communication constraints," in *Second Multi-Robot Systems Workshop*, p. 267278, 2003.

[14] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams: Research articles," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 111–131, 2008.

[15] G. Notarstefano, K. Savla, F. Bullo, , and A. Jadbabaie, "Maintaining limited-range connectivity among second-order agents," in *American Control Conference*, pp. 2124–2129, June 2006.

[16] D. P. Spanos and R. M. Murray, "Motion planning with wireless network constraints," in *American Control Conference*, pp. 87–92, June 2005.

[17] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 313–326, June 1992.

[18] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," vol. 13, no. 3, pp. 398–410, 1997.

[19] K. D. Do, "Formation tracking control of unicycle-type mobile robots with limited sensing ranges," *IEEE Trans. Control Systems Technology*, vol. 16, no. 3, pp. 527–538, 2008.

[20] R. Sharma and S. Hutchinson, "Motion perceptibility and its application to active vision-based servo control," *IEEE Trans. Robot. Automat.*, vol. 13, no. 4, pp. 607–617, 1997.

978-1-4244-2677-5/08/$25.00 ©2008 IEEE