Smart Resource Reconfiguration by Exploiting Dynamics in Perceptual Tasks*

Deepak Karuppiah, Roderic Grupen, Allen Hanson and Edward Riseman
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
{deepak,grupen,hanson,riseman}@cs.umass.edu

***Abstract*— In robot and sensor networks, one of the key challenges is to decide when and where to deploy sensory resources to gather information of optimal value. The problem is essentially one of planning, scheduling and controlling the sensors in the network to acquire data from an environment that is constantly varying. The dynamic nature of the problem precludes the use of traditional rule-based strategies that can handle only quasi-static context changes. Automatic context derivation procedures are thus essential for providing fault recovery and fault pre-emption in such systems.**

We posit that the quality of a sensor network configuration depends on sensor coverage and geometry, sensor allocation policies, and the dynamic processes in the environment. In this paper, we show how these factors can be manipulated in an adaptive framework for robust run-time resource management. We demonstrate our ideas in a people tracking application using a network of multiple cameras. The task specification for our multi-camera network is one of allocating a camera pair that can best localize a human subject given the current context. The system automatically derives policies for switching between camera pairs that enable robust tracking while being attentive to performance measures. Our approach is unique in that we do not make any *a priori* assumptions about the scene or the activities that take place in the scene. Models of motion dynamics in the scene and the camera network configuration steer the policies to provide robust tracking.

***Index Terms*— Sensor networks, distributed perception, resource management.**

I. Introduction

Sensor and actuator networks are gaining increasing attention in cross disciplinary research in robotics and networking due to promising real-world applications that include environmental monitoring [9], robotic swarms [6], agricultural management [2], elderly care [12], habitat monitoring [8], homeland security and transportation [7].

Suppose a future sensing-network company is faced with the challenge of providing a deployment and operating solution to one such application. It must first decide on a configuration for the stationary sensors that provides maximal coverage while using minimal resources. Then it must come up with a strategy to plan, schedule and control the sensors in the network in order to acquire data from the dynamic environment. These two design specifications depart significantly from the current emphasis in sensor network research [1] that emphasizes system level issues, such as communication [5], power consumption, miniaturization, scalability [9], and security [3]. The complementary technology presented in this paper is necessary for practical and deployable sensor networks.

For a sensor network system to deliver on its promises, issues related to information acquisition as well as processing need more attention. Run-time failures can include node crashes and link failures, but also degradation in the quality of observed data or even blackouts caused by inadequate sensor coverage in high interest regions. Further, offline hand-coding of fixed courses of action doesn't respond well to changing situations. Automatic context derivation procedures are essential to deploy scalable information gathering networks. In this paper, we develop ideas based on these observations that lead to automatic context derivation and resource management policies in sensor networks. We shall demonstrate the effectiveness of our approach in a people tracking application using a network of multiple cameras.

Research on multi-sensor networks devoted to human tracking and identification that are closely related to this paper can be found in [11], [10] and [13]. An integrated system has been proposed in [14] for human tracking and face recognition using an active camera network. In [4], a practical distributed vision system based on dynamic memory has been presented. However, in most of the current systems the emphasis is on algorithms that are designed to function only in a preconfigured network configuration. Important issues concerning sensor reconfiguration in the event of failures to maintain information quality is not addressed.

We present a novel strategy for managing redundant camera resources using policies that are derived automatically from models of the environment and camera configurations. For example, when an object enters a region that is nearly directly between two cameras, the triangulation quality deteriorates predictably and can be addressed by switching to a different camera pair [15]. For a given multi-camera configuration (see Fig. 1), we can determine the pair that maximizes static triangulation quality offline. Such a determination yields a context switching policy provided that the environment is stationary. Dynamic environments are more difficult, in general.

In Sections II & III, we define utility of a camera pair based

*This work is supported in part by NASA grant NAG9-01445 and DARPA project DABT63-00-1-0004.

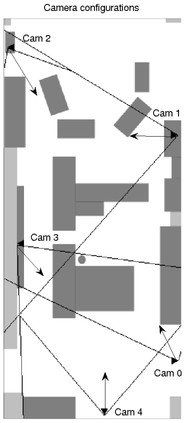


Fig. 1. Floorplan of our lab showing the camera configurations

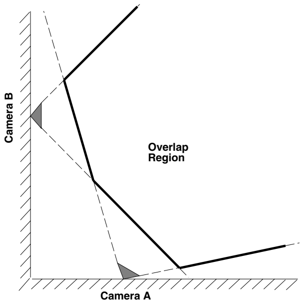


Fig. 2. Binocular FOV for a camera pair R_{AB}

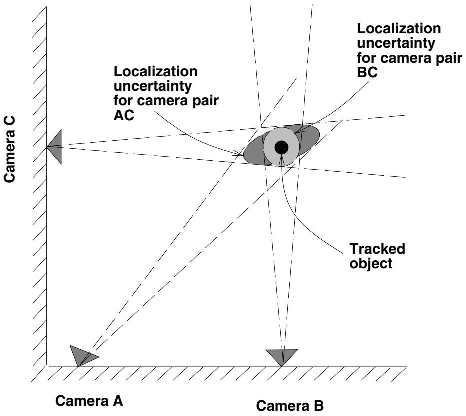

Localization uncertainty for camera pair BC
Localization uncertainty for camera pair AC
Camera C
Tracked object
Camera A
Camera B

Fig. 3. Instantaneous uncertainty in triangulating a target for a kinematic configuration of two camera pairs.

on both static and dynamic factors: including (i) the environment; (ii) motion patterns observed by the pair frequently; (iii) camera pair geometry; and (iv) the current state of the subject. In Section IV, we construct camera pair switching policies using the utility of camera pairs under different run-time contexts. Finally, we shall show the superiority of a reconfiguration policy that exploits inherent characteristics of the task (in this case the dynamics of the moving subject) over an offline policy that solely relies on camera pair geometry.

II. Stationary factors that influence a camera pair's utility

Given a fixed stereo geometry, observational utility over areas of environment can be determined offline in terms of two properties: (1) their binocular field of view (FOV); and (2) the expected precision of triangulation. Within the binocular FOV, the precision of multi-camera localization varies as a function of sensor configuration.

A. Binocular FOV for a camera pair

The configuration of a camera is specified by its location, orientation and field of view. Given this information for both the cameras in a pair, we can easily determine their binocular FOV (Fig. 2). We represent this region for a camera pair, R_{ij}, as a uniform probability density, $\boldsymbol{\kappa}_o$, over $\boldsymbol{p}$ (which are centers of discretized grid cells) as follows :

$$\boldsymbol{\kappa}_o(\boldsymbol{p}, R_{ij}) = \begin{cases} \epsilon & \text{if } \boldsymbol{p} \text{ is within the overlap region} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where ϵ is $1/A$ and A is the total area of the region of overlap. The camera pair R_{ij} may continue to be used as long as its $\boldsymbol{\kappa}_o$ measure is non-zero.

B. Quality of triangulation

We use a localizability metric to evaluate the triangulation quality of different camera pairs. The localizability metric is instantaneous uncertainty in the location of a subject. Fig. 3 illustrates the difference in localization precision for two pairs of cameras. Camera pair R_{BC} is better suited to localize the subject instantaneously.

If B is the baseline between two cameras and θ_1 and θ_2 are the respective headings to the subject, the uncertainty Jacobian is given as follows :

$$J = \frac{B}{\sin^2(\theta_2 - \theta_1)} \begin{pmatrix} \cos\theta_2 \sin\theta_2 & -\cos\theta_1 \sin\theta_1 \\ \sin^2\theta_2 & -\sin^2\theta_1 \end{pmatrix} \tag{2}$$

The instantaneous localizability measure $\boldsymbol{\kappa}_p$ is defined as $\sqrt{|JJ^T|}$, where $|\cdot|$ is the determinant of a matrix. The lower this measure is, the higher the precision of localization. Depending on the cost associated with switching to a different pair, the current pair may continue to be used as long as the uncertainty is within acceptable limits.

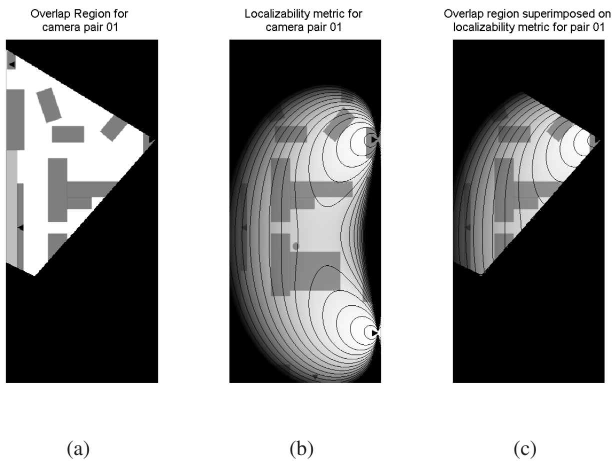
Since the precision as defined above is determined only by the kinematic configuration of the cameras and a location, we shall represent the triangulation quality measure as $\boldsymbol{\kappa}_p(\boldsymbol{p}, R_{ij})$.

We can compose both the stationary measures defined above to get

$$\boldsymbol{\kappa}(\boldsymbol{p}, R_{ij}) = \boldsymbol{\kappa}_o(\boldsymbol{p}, R_{ij}) \cdot \boldsymbol{\kappa}_p(\boldsymbol{p}, R_{ij}) \tag{3}$$

Fig. 4 shows the composition of $\boldsymbol{\kappa}_o$ and $\boldsymbol{\kappa}_p$ for a camera pair in our environment.

Thus, for every location in the room, we can determine *a priori* the camera pair that is best suited to localize the


Overlap Region for camera pair 01
Localizability metric for camera pair 01
Overlap region superimposed on localizability metric for pair 01

(a) (b) (c)

Fig. 4. Composition of $\boldsymbol{\kappa}_o$ and $\boldsymbol{\kappa}_p$ for camera-pair R_{01}. The gray regions represent stationary objects in the room. The dark regions indicate high uncertainty in triangulation while bright regions indicate low uncertainty. The contours indicate regions of similar uncertainty. The black regions are invalid for this camera pair and hence events happening there cannot be observed by it.

stationary object in those regions.

$$R^*(\boldsymbol{p}) = \arg\min_{R_{ij}} \boldsymbol{\kappa}(\boldsymbol{p}, R_{ij}) \tag{4}$$

III. Dynamic factors that influence a camera pair's utility

Aggregation of commonly observed track trajectories provides a wealth of information about the camera pair's utility that is distinct from the offline utility described in the previous section. For example, such models could predict: 1) where activities occur (and by implication where they don't or where they are unobservable); 2) where they occur most frequently; and 3) the correlation between activities occurring at one place with another. Such predictive models are very useful in planning allocation of cameras to monitor activities because resources are not wasted by looking at regions where there is very sparse or no activity. For instance, activities observed along narrow pathways can be observed better by cameras directed along those paths. Further, alternate pairs that are best suited to take over the tracking can also be identified.

In the following, the utility of a camera pair is modelled as a likelihood density abstracted from observed track trajectories.

A. Activity density for a camera pair

Comparing Fig. 2 with Fig. 5, we see that the overlap region is not solely dependent on camera configurations. The presence of a single stationary object (like a partition) can carve up the space in complex ways. While some objects may create occlusion regions, others may not have any effect at all. Obviously, when the camera network has to cover large areas, it is a daunting task to model every occluding surface for all camera pairs.

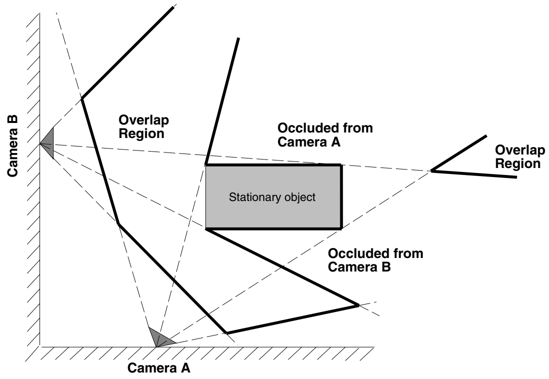


Fig. 5. Region of overlap for a camera-pair affected by objects in the environment

Our strategy is, therefore, to observe activities in the scene over a long duration and record the camera pairs that observed an activity in a particular location. This helps us to model the activity density for a camera pair and such a model can in fact closely approximate the true overlap region. Although we propose to model the occluding boundaries in 2-D, our strategy can easily be extended to produce a 3-D version that would be more accurate and useful. Fig. 6 illustrates the process of abstracting the density of activity in different grid locations from long term observations made by a camera pair. We use kernel density estimation to model the activity density. The probability of observing motion (or activity) at a grid location $\boldsymbol{p}$ using a resource R_{ij} is given by

$$\boldsymbol{\xi}_{ij}(\boldsymbol{p}) \propto \sum_{k=1}^{N_{ij}} K_h(\boldsymbol{p} - \boldsymbol{p}_k) \tag{5}$$

where $\boldsymbol{p}_k$ are the locations where motion was observed, N_{ij} is the total number of observations made by R_{ij} and $K_h(\cdot)$ is a suitable kernel function with a bandwidth parameter h.

Now, the overall probability of observing motion at a grid location from all resources is given by

$$\boldsymbol{\xi}(\boldsymbol{p}) = \sum_{j=1}^{N} K_h(\boldsymbol{p} - \boldsymbol{p}_j) \tag{6}$$

where $N = \sum_i N_i$ is the total number of observations made. Finally, the likelihood of observing an activity at a grid

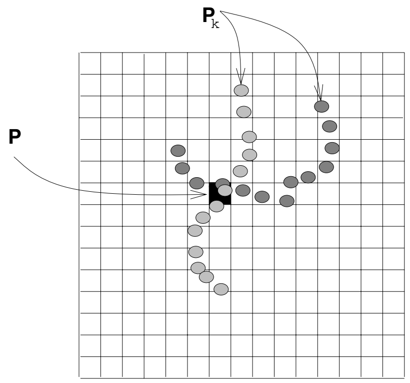


Fig. 6. Modelling activity density using uniform kernel functions. For each location P, the past observations P_k in its neighbourhood are used to estimate the probability of activity there.

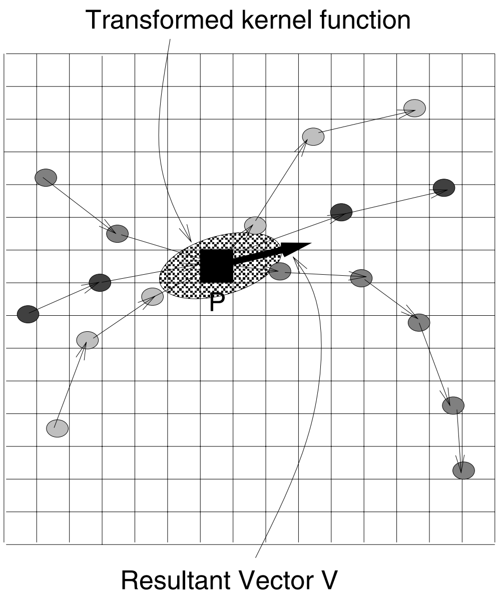


Fig. 7. Modelling activity density using deformed kernel functions based on locally dominant motion vectors

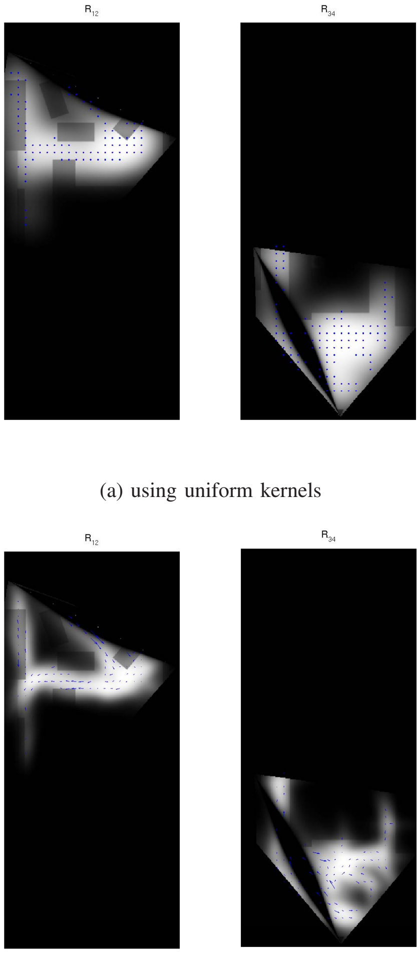
location using a camera pair is given by

$$\begin{aligned}\boldsymbol{\nu}(\boldsymbol{p}, R_{ij}) &= \Pr(\boldsymbol{p}|R_{ij}) \\ &= \frac{\boldsymbol{\xi}_i(\boldsymbol{p})}{\boldsymbol{\xi}(\boldsymbol{p})} \cdot \boldsymbol{\kappa}(\boldsymbol{p}, R_{ij}). \qquad (7)\end{aligned}$$

Note that, if $\boldsymbol{p}$ is in the invalid region of the cameras in R_{ij} as determined by the offline utility measure $\boldsymbol{\kappa}$, then $\boldsymbol{\nu}(\boldsymbol{p}, R_{ij})$ is zero.

B. Activity density biased by trajectory dynamics

The discussion thus far has ignored the trajectory dynamics while evaluating the occupancy grid density from the observed data. The dynamic patterns observed most frequently by each camera pair can also inform the pair's utility. We propose to take advantage of this by doing anisotropic kernel density estimation. Fig. 7 shows how the bandwidth parameter of a kernel is modified for a location $\boldsymbol{p}$ in the grid. For each location $\boldsymbol{p}$ in the grid, we first compute the resultant $\vec{V}(\boldsymbol{p})$ of all the motion vectors passing through that grid. The uniform kernel K_h is scaled and rotated using the magnitude $|\vec{V}|$ and the direction θ of the resultant vector respectively. The transformed kernel function $K_{|V|,\theta}$ is then used to estimate the kernel density in (5) & (6). The modified likelihood $\boldsymbol{\nu}'$ is evaluated in the same way as in (7). Fig. 8 compares the estimation of $\boldsymbol{\nu}(\cdot, R_{12})$ using uniform and anisotropic kernels. We can see that uniform kernels tend to introduce significant likelihood even in regions of infrequent activity because they smooth uniformly in all directions. The anisotropic kernels produce likelihood density that is biased heavily along directions of dominant motion. Thus the utility of the camera pair remains high as long as the tracked object follows a direction of dominant motion that was observed frequently by the camera in the past.



(a) using uniform kernels

(b) using anisotropic kernels

Fig. 8. Estimation of activity density using camera pairs R_{12} and R_{34}. (a) Uniform kernels tend to introduce significant likelihood even in regions of infrequent activity because they smooth uniformly in all directions. (b) The anisotropic kernels produce likelihood density that is biased heavily along directions of dominant motion.

IV. Context-switching policies

Our ultimate goal is to construct useful camera pair switching policies. Such policies can either rely on the stationary factors (Section II) alone or include the dynamic environmental factors (Section III) in addition to the stationary factors. Therefore, we have two variations of policies for choosing the best camera pair - one that exploits the motion dynamics of the tracked object (D-policy) and one that doesn't (Q-Policy). We shall now present the two policies, compare their performances and discuss their relative trade-offs.

Let $\bar{R} \equiv \{R_{ij}\}$ be the set of all camera pairs available to track a moving object. Let the position of the tracked object at time t be denoted by $\boldsymbol{p}(t)$. The position vector may also be augmented with the velocity vector, $\dot{\boldsymbol{p}}$, to take into account the object's motion.

A. Static switching policy : Q-Policy

The track quality for a camera-pair R_{ij} given the current position estimate $\boldsymbol{p}(t)$ is computed using the Bayes' rule :

$$\Pr(R_{ij}|\boldsymbol{p}(t)) \propto \Pr(\boldsymbol{p}(t)|R_{ij}) \cdot \Pr(R_{ij}). \qquad (8)$$

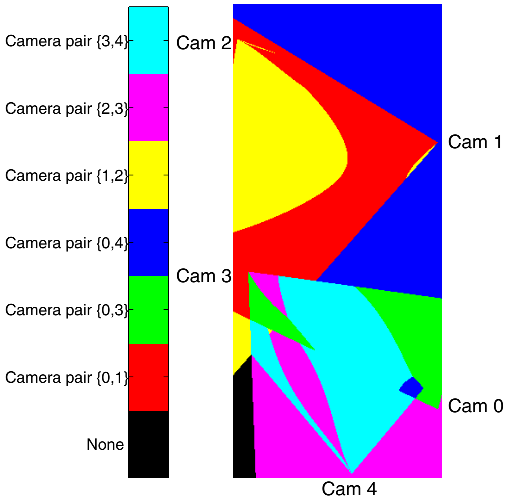


Fig. 9. Each color indicates a camera pair to be used to track in that location. The color to camera pair mapping is shown on the colorbar to the left. The locations that are black indicate that those regions are unobservable.

The first probability term on the right hand side of the above equation is $\boldsymbol{\kappa}(\boldsymbol{p}(t), R_{ij})$ (from (3)). The second term can be ignored because all the camera-pairs are equally likely under the assumption of stationarity.

Finally, the maximum *a priori* estimate of (8) gives us the best resource,

$$R^* = \arg\max_{ij} \Pr(R_{ij}|\boldsymbol{p}(t)), \tag{9}$$

that should be used to track the object given the current context.

Thus, Q-Policy is essentially a table-lookup in which, given the instantaneous state of the tracked object (estimated by the current camera pair), the corresponding optimal camera pair is chosen during the next time-step. Q-Policy is essentially greedy on the triangulation quality. Fig. 9 shows the result of the choosing the best camera pair using Q-Policy.

B. Dynamic switching policy : D-Policy

To account for environmental factors, the first probability term on the right hand side of (8) is replaced by $\boldsymbol{\nu}'(\boldsymbol{p}(t), R_{ij})$ (from (7)). As we have seen in Section III, $\boldsymbol{\nu}'$ is influenced by the expected observability of a state with the camera pair R_{ij} as well as the expected localization uncertainty and region of overlap of R_{ij} at that state.

The second term of (8) is the fraction of times R_{ij} reported a valid observation. This term is the prior probability of a camera pair determined by considering the long-term participation of a camera pair in tracking the activities.

Finally, this policy takes the dynamic motion of the object into account. The choice of camera pair proceeds according to the algorithm shown in Algorithm 1. D-Policy minimizes the number of context switches by allowing for some degradation in the triangulation quality.

Algorithm 1 D-Policy

```
$t \leftarrow 1$
$R \leftarrow QPolicy(\boldsymbol{p}(t))$
loop
    $\boldsymbol{p}^*(t) \leftarrow PREDICT(\boldsymbol{p}(t))$ {predict state}
    $R_{new} = QPolicy(\boldsymbol{p}^*(t))$
    if $R_{new} \neq NONE$ then {object not in blind zone}
        if $\boldsymbol{\nu}'(\boldsymbol{p}^*(t), R) < Threshold$ then {only when utility below Threshold}
            $R \leftarrow R_{new}$ {switch camera-pair}
        end if
    end if
    $t \leftarrow t + 1$
    $\boldsymbol{p}(t) \leftarrow ESTIMATE(R)$ {estimate state using observation from camera-pair R}
end loop
```

C. Performance comparison of the policies

We compare the performance of the two camera pair switching policies described above by evaluating a few performance measures. The evaluation depends entirely on the user's preferences. For example, they can be evaluated based on (i) the total cost of context switches, (ii) tracking uncertainty and (iii) time to fail. It is generally desirable to keep the frequency of switching between camera pairs to a minimum as constant switching leads to instability in the tracking process itself. However, when tracking uncertainty must be kept below a certain threshold, frequent switching may become unavoidable. Similarly, the system should not wait for a failure to occur before switching resources. Incorporating dynamics can help predict such failures and consequently increase the average time to fail. The overall character of the system can thus be defined by carefully weighting these factors relative to one another.

V. Experimental Results

Our experimental smart space facility consists of five Sony EVI-D100 PTZ cameras mounted on walls (see Fig. 1). A compute rack consisting of a cluster of ten VMIC single board computers each with a 928 Mhz processor and 256 MB RAM forms the computational backbone for the cameras. The nodes in the cluster share a 100Mbps and a 1000Mbps Ethernet link. Each node has a Leutron vision frame grabber to which a camera is connected. Each node acts as a server of raw video as well as processed information from the cameras.

The cameras are organized into camera pairs, each pair capable of localizing a target in the environment. The cameras are calibrated and using their stationary configurations we determine the uncertainty and overlap region density described in Section II. We observed typical activities in the room for 5 different episodes each for a duration of 1 hour using all possible camera pairs. From this data, we model the likelihood densities described in Section III.

The results of the two policies are shown for 5 typical paths in Fig. 10. The performance evaluation based on the number

TABLE I

EVALUATION OF Q-POLICY (FOR PATHS SHOWN IN FIG. 10)

Path No.	No. of switches	Steps per context switch
1	5	19.50
2	7	14.13
3	18	7.37
4	12	10.31
5	5	19.83

TABLE II

EVALUATION OF D-POLICY (FOR PATHS SHOWN IN FIG. 10)

Path No.	No. of switches	Steps per context switch
1	3	28.75
2	2	36.00
3	5	21.17
4	3	31.25
5	2	38.67

TABLE III

MEAN DIFFERENCE IN THE LOCALIZABILITY METRIC BETWEEN D-POLICY AND Q-POLICY

Path No.	1	2	3	4	5
Difference	0.1471	0.1343	0.0576	0.1122	0.1151

of context switches, average path length per context switch and the triangulation quality trade-off between the policies is shown in Tables I, II & III.

We can observe that the D-Policy indeed reduces the frequency of context switches. When the current dynamics of the tracked object is supported adequately by the current camera pair's activity density model, it is better off using the same resource instead of initiating the context switch. Whereas in Q-Policy, the preoccupation is more with precision of localization and thus costly context switches are not prejudicial. As a result, we see a pattern of frequent context switches there. Since by design, D-Policy cannot perform better than Q-Policy in terms of triangulation quality, we see that this is being traded-off for reduction in the number of context switches.

VI. CONCLUSION

Sensor network research has thus far been focussed on systems level issues. Though these issues are vital, for sensor networks to really grow into real-world systems emphasis must include characterising and maximizing the quality of information gathered by individual sensors or subsets of sensors.

Traditional design approaches cannot be applied to sensor networks to accomodate this new emphasis because run-time contexts tend to be driven by the constantly changing environment. Thus, a complete specification that includes all input data sets is impossible *a priori* and techniques that optimize on-line are required.

In this paper, we propose a strategy for sensor targeting that is aware of the information generating processes in the environment. We have shown an automatic and scalable technique for deriving context-switching policies for a widely relevant application, that of tracking human subjects using multiple cameras. By observing human activities over extended periods of time, we model the routinely occurring events and rank the camera pairs that maintain fault-free tracking over different regions of the room. Camera pairs are also ranked based on stationary factors such as region of overlap and triangulation quality for different regions of the scene. Finally, we have derived two policies and have shown that the D-Policy minimizes the number of context switches by allowing for some degradation in the triangulation quality when compared to the Q-Policy.

One can observe that the placement of cameras in our room does not provide adequate coverage of all the activities in the room. In future, we plan to design an optimization technique that would generate an optimal camera network configuration. Such a network would provide maximal coverage with minimum cameras while guaranteeing a lower bound on expected triangulation quality for all regions of the room.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[2] Jenna Burrell, Tim Brooke, and Richard Beckwith. Vineyard computing: sensor networks in agricultural production. *Pervasive Computing*, pages 38–46, January-March 2004.

[3] Laurent Eschenauer and Virgil D Gligor. A key-management scheme for distributed sensor networks. In *CCS'02*, Washington DC, USA, November 18-22 2002.

[4] T. Matsuyama et al. Dynamic memory: Architecture for real time integration of visual perception, camera action, and network communication. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 728–737, Hilton Head Island, SC, June 2000.

[5] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *Mobile Computing and Communications Review*, 5(4):10–24, 2002.

[6] Vijay Kumar, Daniela Rus, and Sanjiv Singh. Robot and sensor networks for first responders. *Pervasive Computing*, 3(4):24–34, Oct-Dec 2004.

[7] Dan Li, Kerry Wong, Yu Hen Hu, and Akbar Sayeed. Detection, classification and tracking of targets in distributed sensor networks. *IEEE Signal Processing Magazine*, 19(2):1–23, March 2002.

[8] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *WSNA*, pages 88–97, 2002.

[9] Kirk Martinez, Jane K. Hart, and Royan Ong. Environmental sensor networks. *Computer*, 37(8):50–56, August 2004.

[10] A. Nakazawa, H. Kato, and S. Inokuchi. Human tracking using distributed vision systems. In *14th International Conference on Pattern Recognition*, pages 593–596, Brisbane, Australia, 1998.

[11] Kim C. Ng, H. Ishiguro, Mohan M. Trivedi, and T. Sogo. Monitoring dynamically changing environments by ubiquitous vision system. In *IEEE Workshop on Visual Surveillance*, Fort Collins, Colorado, June 1999.

[12] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron. Monitoring behaviour in home using a smart fall sensor. In *IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, pages 607–610, October 2000.

[13] A. Pentland. Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):107–119, January 2000.

[14] Mohan M. Trivedi, K. Huang, and I. Mikic. Intelligent environments and active camera networks. *IEEE Transactions on Systems, Man and Cybernetics*, October 2000.

[15] Z. Zhigang, D. R. Karuppiah, E. M. Riseman, and A. R. Hanson. Keeping smart, omnidirectional eyes on you. *Robotics and Automation Magazine: Special Issue on "Panoramic Robots"*, 11(4):69–78, December 2004.

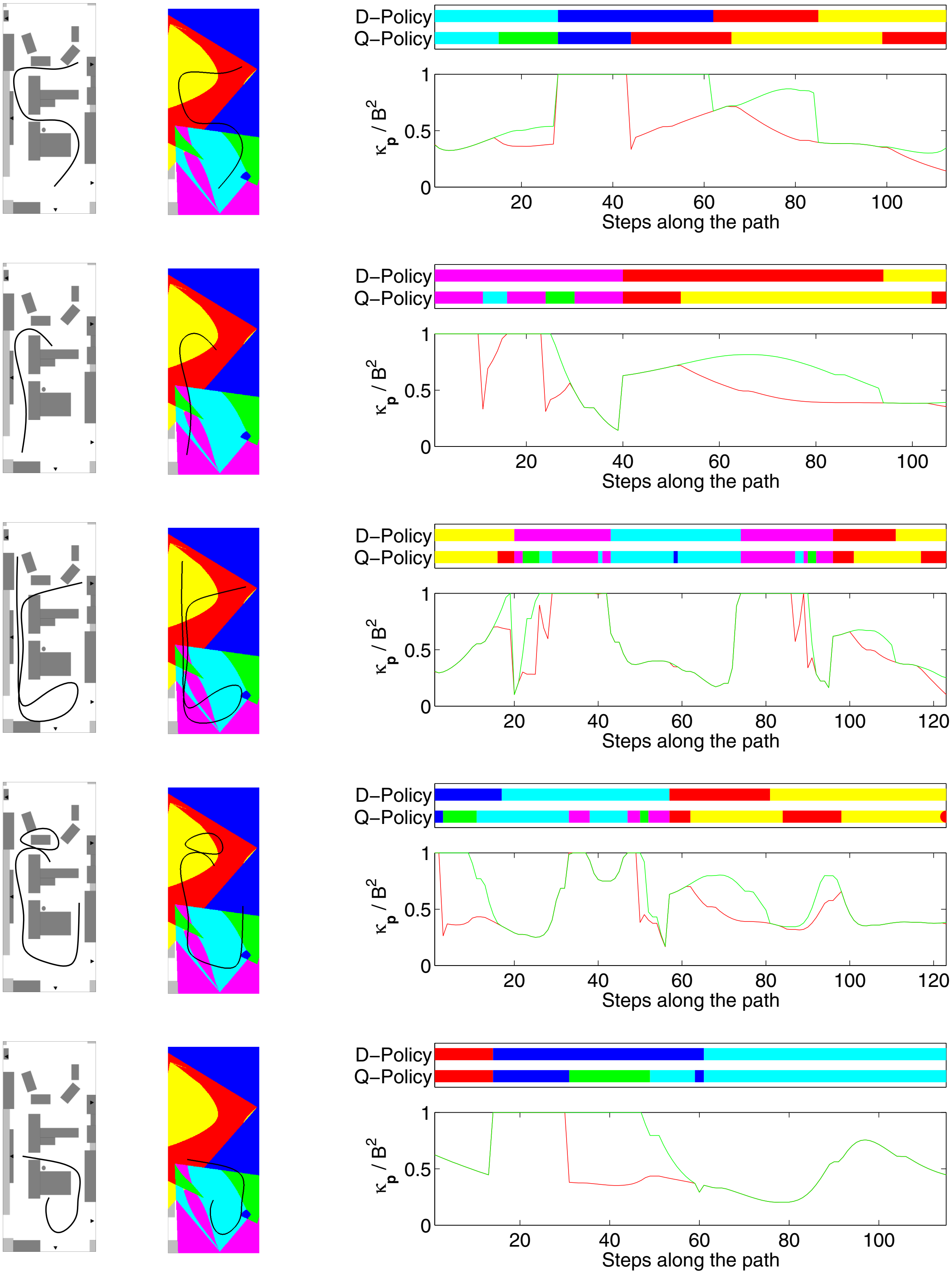


Fig. 10. Comparison between Q-Policy and D-Policy. The first two columns show the paths overlaid on the floorplan of our lab and the color coded map of Q-Policy respectively. The last column shows the normalized localizability metric along the path for both Q-Policy(red) and D-Policy(green) along with their context switches. Note that while the Q-policy favors greedy context switches to minimize the triangulation error, the D-Policy tolerates some degradation in triangulation error for fewer context switches