Fusion of Multiple Trackers in Video Networks

Yiming Li
Center for Research in Intelligent Systems
University of California at Riverside
Riverside, CA, USA
yimli@ee.ucr.edu

Bir Bhanu
Center for Research in Intelligent Systems
University of California at Riverside
Riverside, CA, USA
bhanu@cris.ucr.edu

***Abstract*— In this paper, we address the camera selection problem by fusing the performance of multiple trackers. Currently, all the camera selection/hand-off approaches largely depend on the performance of the tracker deployed to decide when to hand-off from one camera to another. However, a slight inaccuracy of the tracker may pass the wrong information to the system such that the wrong camera may be selected and error may be propagated. We present a novel approach to use multiple state-of-the-art trackers based on different features and principles to generate multiple hypotheses and fuse the performance of multiple trackers for camera selection. The proposed approach has very low computational overhead and can achieve real-time performance. We perform experiments with different numbers of cameras and persons on different datasets to show the superior results of the proposed approach. We also compare results with a single tracker to show the merits of integrating results from multiple trackers.**

Keywords- fusion, camera selection, tracking

I. INTRODUCTION

Many tasks of modern video surveillance systems highly depend on the tracking results for objects. For example, when following a suspect at an airport which is covered by hundreds of cameras, we want to display a group of images from cameras with related views of the person on the monitor wall in a control room. The cameras can hand-off this person from one to another as the suspect walks/runs around. We call this process *camera selection*. In the ideal case a tracking algorithm runs on each camera and it broadcasts the status of following this person in the network. Thereafter, a decision whether or not to display the image of a camera is made by each camera locally or by a central controller. Unfortunately, in an imperfect world, if the tracking results returned by the tracking algorithm are wrong or being more and more inaccurate, the camera hand-off decision and the display decision will be misleading and the suspect will be finally lost.

There are many trackers available thanks to the research in the past three decades. In the real-world scenarios, it is hard for any tracker to track robustly under all the situations, not matter how sophisticated the tracker is. On the other hand, there can always be new trackers coming out with a better performance under one or more specific conditions. Our goal in this paper is to find a generic way to make use of the advantages of existing trackers to make tracking more reliable, especially for performing camera selections in a video network. In the meantime, it will not be hard to integrate new trackers to the proposed system. We propose a score-level fusion of multiple trackers with consideration of camera selection quality as well.

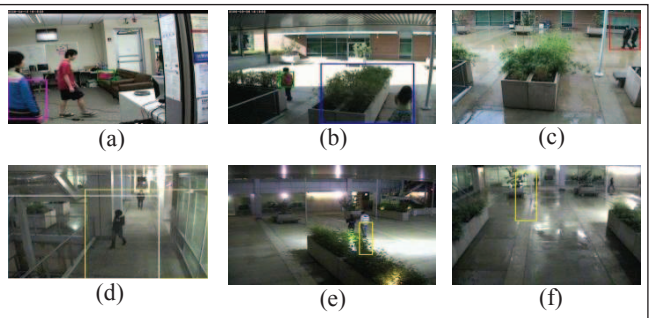

Figure 1. Some error that will propagate and influence camera selection results. (a) the Camshift tracker for the person in red is distracted to the red pillow. (b) The particle filter tracker cannot distinguish the person and the plants. (c) The CamShift tracker cannot distinguish two persons whose clothes are in similar color. (d) Both the particle filter and the CamShift tracker cannot handle the bad illumination condition. (e) and (f) The on-line boosting tracker loses the person as the person goes too far away from the camera.

II. RELATED WORK AND OUR CONTRIBUTIONS

Many trackers, such as the CamShift tracker [1], the particle filter tracker [2], a series of on-line boosting trackers [3][4][5], etc., have been proposed during the past decades. These trackers either are updated according to some dynamic stochastic processe or treat tracking as a classification problem. There are also some trackers which fuse different types of sensors, e.g., in [6], the authors use both audio and video sensors. In [7], the authors use infrared cameras together with video cameras. Different types of trackers may achieve different performances under different application scenarios because of their inherent properties. For example, the CamShift tracker [1] is very simple such that it can be used to track videos with very high frame rates, the particle filter tracker [2] is well suited for occlusions and the series of on-line boosting trackers [3][4][5] are less sensitive to poor illumination conditions.

However, none of these trackers is capable of dealing with all kinds of circumstances, especially when it comes to a long period of time. Thus, a slight inaccuracy occurred in the tracker may lead to a wrong camera selection decision by the

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

978-1-4577-1707-9/11/$26.00 ©2011 IEEE

system. This error may propagate such that system finally loses the track of one or more objects. Some explains are shown in Figure 1. In this paper, we propose an approach that fuses the performance of multiple trackers such that we can make better use of the information returned by the tracker with higher confidence when decide which camera to use. This is different from the approaches which do feature-level [3] or sensor-level [6][7] fusions for a single tracker to enhance the performance of a single tracker. What we propose in this paper is to do a system-level fusion of multiple trackers to bring a more satisfying camera selection/hand-off solution.

The contributions of this paper are: 1) We propose an approach which does a score-level fusion of multiple trackers for solving the camera selection/hand-off problem. 2) We use several state-of-the-art trackers to do the experiments and have a discussion on the optimal number of trackers to be used. We do real-time experiments with real-life data under different circumstances to evaluate the efficiency and robustness of the proposed approach and compare that with other camera selection approaches without the fusion of multiple trackers.

III. Camera Selection with Fusion of Multiple Trackers

In this section, we will present the idea of doing camera selection/hand-off based on the fusion of multiple trackers. We first list the symbols and notations to be used in the rest of this paper in Table I. The basic flow of the algorithm is illustrated in Figure 2.

TABLE I. Symbols and Notations

Symbols	Notations
N_P	Number of persons in the system
N_C	Number of cameras in the system
N_T	Number of trackers used for each person
P_i	Person i
C_j	Camera j
$n_C(i)$	Number of cameras that can see P_i
$score^x_{track}$	Score to evaluate the tracking quality for tracker x
$score^x_{camSel}$	Score to evaluate the camera selection quality in tracker x
α	Parameter deciding the speed of memory fading
λ	Parameter giving penalty to the current tracker considering the tracking quality of other trackers
$Crt^x(i)$	Criterion i's value for tracker x
N_{Crt}	Number of criteria

A. Fusion of Multiple Trackers

Assume that for a particular person P_i, there are $n_C(i)$ cameras that can see this person. Suppose we have N_T trackers which run on all cameras. Thus, for each particular P_i, there are $n_C(i) \times N_T$ tracking results all together. We call them hypotheses. For each tracker $x, x\epsilon\{1, \dots, N_T\}$, we calculate its associated tracking score $score^x_{track}$ and the camera selection score $score^x_{camSel}$. The final score for each tracker, based on which we do the final camera selection, is as given below:

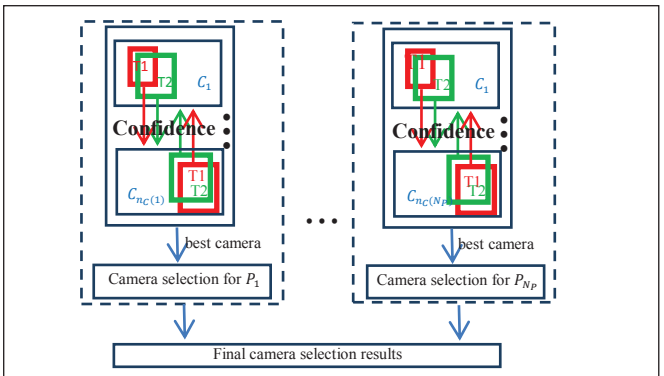

Figure 2. Basic flow of the algorithm. In this figure, it is supposed that we use two trackers, T_1 and T_2, for each detected person. The confidence of each tracker is broadcasted in the system. The best camera is then selected for each person based on the final score calculated from the confidences as described in Equations (1) – (4).

$$score^x = \alpha \cdot score^x_{track} + (1-\alpha) \sum_{i=f-m}^{f} (score^x_{track}(i))^\alpha + \lambda \cdot score^x_{camSel} \quad (1)$$

$$score^x_{track} = conf^x \quad (2)$$

$$\lambda = \frac{e^{conf^x}}{\sum_{x=1}^{N_T} e^{conf^x}} \quad (3)$$

$$score^x_{camSel} = \sum_{j=1}^{N_{Crt}} Crt^x(i) \quad (4)$$

In Equation (1), $score^x$ is made up of two parts: 1) The tracking quality of the current tracker x, which has a fading memory of its performance from the previous m frames up to the current frame. That is, each tracker exponentially discounts the influence of its past tracking quality in the computation of its current tracking quality. The parameter α controls how fast we want the memory to fade away. This formulation allows us to consider the performance of a tracker continuously, such that when we consider to hand off from one camera to another, the temporal smoothness is also taken into account. 2) The camera selection score, which has a penalty weight decided by the performance of other trackers that can see the same person. It is easy to predict from Equation (1) that if there is other tracker with higher tracking confidence, then the camera selection score of the current tracker is downgraded. This is the part where we actually fuse the performance of multiple trackers. Traditional camera selection/hand-off approaches are only based on a single tracker. However, in real applications, although the tracking result is satisfying for a tracker, i.e., it is not too far away from the tracked person, sometimes it is not accurate enough to provide the information which is needed by the camera selection/hand-off approaches. By applying the proposed idea, the final camera selection result relies more on the information returned by the tracker with higher confidence and, thus, reduces the uncertainty of the camera selection/handoff procedure. This is illustrated in Figure 3. Assume we are only using the Camshift tracker (CS, solid line)

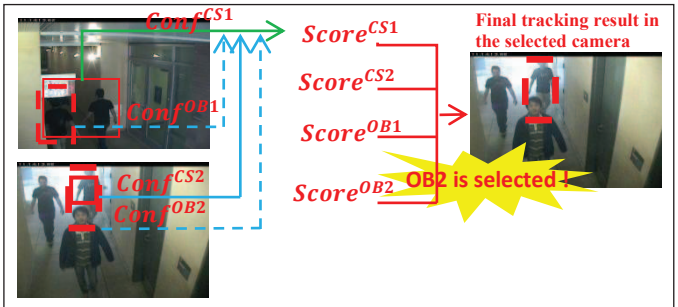

Figure 3. Illustration of fusing multiple trackers.

and the on-line boosting tracker (OB, dashed line). The thickness of a bounding box implies its confidence in tracking this person. As we can see, when deciding the $score^{CS1}$, the confidences of the CS tracker (implied by the green line) and all the other trackers (implied by the blue lines) on the same person are simultaneously considered. Thus, although the camera selection score for CS1, $score^{CS1}_{camSel}$, is higher than any other tracker, the system still choose the OB2 for this person, which has a higher tracking quality.

We calculate the tracking score of the current tracker as the confidence value that returned by it. In our experiments, we implement two categories of trackers based on the different features they use: 1) the CamShift tracker (CS) and the particle filter tracker (PF) which use HSI color as the feature; 2) the on-line boosting tracker (OB), semi-supervised on-line boosting tracker (SOB) and the multiple instance learning tracker (MIL), which use a feature pool consisting of histogram of orientations, Haar wavelets and local binary patterns (LBP). For the first category of trackers, we calculate the tracker confidence as the correlation coefficient of the color histogram of the person's bounding box returned by the tracker between the current frame and the previous frame multiplied by the previous frame's confidence. For the second category of trackers, we use the confidence returned by the boosting algorithm, which is a weighted summation of a group of weak classifiers, as the tracker confidence (for more information on the calculation, the readers can refer to [3]).

The camera selection score $score^{x}_{camSel}$ is based on the user-supplied criteria for doing camera selections. In our experiments, we apply the same criteria as those in [8] (size, position and view of the object) plus region covariance [9] and a spatial smoothness criterion for intuitive observation and easy comparison. Note that the spatial smoothness refers to the smoothness of the tracks, whereas the temporal smoothness we mentioned previously refers to the usage of a camera to track a person. A camera selection algorithm is temporal smooth means that we do not switch among cameras to track the same person too frequently.

The overall algorithm is described in Algorithm 1.

Algorithm 1. Camera Selection with Fusion of Multiple Trackers

The algorithm is run for each frame t for camera selection.
Input: For each person, the bounding box coordinates $\boldsymbol{X}(i,j,t)$ from all trackers in all the cameras that can see this person.
Output: Camera ID of the camera to track each person.

for i = 1:N_P
{
  for j = 1:$n_C(i)$
  {
    St(t,j) = calcSt($\boldsymbol{X_t}(i,j), St(t-1,j)$);
    Sc(t,j) = calcSc(***X_t*** *(i,j)*);
  }
  Score*(t)* = calcScore(**St**(t), **Sc**(t));
  camID(i,t) = $\arg_j \max$ {$\boldsymbol{Score}(t)$};
}
calcSt($\boldsymbol{X_t}(i,j), St(t-1,j)$
{
  Calculate $score^{x}_{track}$ according to Equation (2).
}
Sc(t,j) = calcSc(***X_t*** *(i,j)*)
{
  Calculate $score^{x}_{camSel}$ according to Equation (4).
}
$\boldsymbol{X_t}(i,j)$ = [$x_{topLeft}(i,j,t)$, $y_{topLeft}(i,j,t)$, $x_{botoomRight}(i,j,t)$, $y_{bottomRight}(i,j,t)$];
St*(t)* = [*St(t, 1),…,St(t,* $n_C(i)$*)*];
Sc*(t)* = [*Sc(t, 1),…,Sc(t,* $n_C(i)$*)*];
Score*(t)* = [*Score(t, 1),…,Score(t,* $n_C(i)$*)*];

IV. Experimental Results

In this section, we describe experimental data set, the environment where experiments take place and compare the results with other approaches for camera selection /hand-off.

A. Data

To show the robustness of the proposed approach, we do our experiments in a physical camera network, which consists of 37 outdoor commercial available Axis 215 IP cameras. The experiments are carried out at different times of a day. We also test the proposed approach on the publicly available datasets PETS2009 S2.L1. Since different trackers may achieve different tracking quality under different illumination conditions, frame rates, extent of occlusions etc., we do comparisons with other approaches to show the stability of the proposed approach due to the fusion of multiple trackers.

B. Trackers Used in the Experiments

We select 5 trackers based on different principles to do our experiments. We use different numbers and combinations of trackers. We run our experiments on a computer with Intel Core 2 Duo 3.16GHz CPU, 4G memory. Each camera is manipulated as a single thread. It turns out that, if we use 2 to 3 trackers for a person, the program can process at least 20 frames per second; if we use 4 trackers for a person, it can process at least 12 frames per second; if 5 trackers are used for a person, it can process around 8 frames per second. So in the following experiments, we choose 3 trackers: PF, SOB and MIL. The reason why we choose these 3 trackers is that we want to have trackers based on different features so that they can compensate each other. We try several different tracker combinations and the PF/SBO/MIL works the best. Different combinations of trackers and their performances for case 1 (see Table III) is listed in Table II. We compare our result with the ground truth (manually labeled by using the ViPER-GT

TABLE II. EXPERIMENTAL CASES

Cases	N_P	N_C	Capture time	Video length(frames)
Case 1	5	4	daytime	896
Case 2	4	4	evening	942
Case 3 (PETS2009 S2.L1)	9	5	daytime	795

ground truth tool), if the overlap of our bounding box and the ground truth bounding box is larger than 70% and the size of our bounding box is less than 1.5 times of the ground truth bounding box, we treat this as a correct tracking. The correct tracking percentage in Table III is the number of correctly tracked frames divided by the number of all the frames in the video sequence (averaged by the number of persons).

When tracking multiple persons, the correspondences of persons are built by using homographies. The homographies of our own data are pre-calculated by providing corresponding laser points in a neat scene at night. The ones for the PETS datasets are calculate by manually picking up corresponding pairs. Because of this reason, only 5 views (view 1, view 5, view 6, view 7 and view 8) of the S2.L1 data are selected, since it is hard to find points in correspondence for the other two views.

TABLE III. TRACKER PERFORMANCES IN CASE 1

Combination of trackers in Case 1	Process speed (fps)	Correct tracking percentage
CS/PF/OB	25	91.73
CS/PF/SOB	25	89.91
CS/PF/MIL	24	90.96
CS/OB/MIL	22	85.97
CS/OB/SOB	23	89.96
CS/SOB/MIL	21	90.02
PF/OB/SOB	21	94.34
PF/OB/MIL	21	95.62
PF/SOB/MIL	21	95.89
OB/SOB/MIL	20	92.81

C. Results and Analysis

1) Experiments on Our Own Datasets

For our own datasets, we collect data for both daytime and evening time. This is because we want to show how differently a tracker performs under different environmental conditions.

Figure 4. Some results for camera selection. The left column shows the case in the day time, tracking the person in yellow clothes. The right column shows the case in the evening, tracking the person in black. We only show images of the cameras which are selected to track the person. The first row is the result when using the particle filter tracker (PF) only; the second row is the result when using the semi-supervised on-line boosting tracker (SOB) only; the third row is the result when using the multiple instance learning tracker (MIL) only; the fourth row is the result when fusing all the above 3 trackers together.

For example, as shown in Figure 4, we can notice that, when the lighting condition is good, the color-based particle filter tracker performs better than the other two, e.g., it gives higher tracking confidence and can deal with most of the occlusions. However, in the evening time, when the lighting condition is poor, the particle filter tracker, which uses color as its feature, almost performs randomly and is much worse than the other two trackers. Similarly, when the tracker uses texture, orientation, etc. as the feature, the trackers performance is gradually downgraded as the size of the object becomes small, in which case a color-based tracker can work better. When we use the proposed approach to fuse multiple trackers, it will be biased to the tracker with higher confidence, since it can consider the impacts from other trackers' tracking confidence when the current tracker has a low tracking quality.

Note that we only show the captured image from the camera which is selected by the system to track a person. For example, in case 1, we deploy 4 cameras, whereas only some of these 4 are selected for the 3 frames that are shown. Different cameras could be selected when applying different trackers. This is because different trackers will provide different bounding boxes (different sizes, positions, etc.), which may lead to different camera selection results. For example in Frame_212, when using the SOB tracker, camera 17 yields a better bounding box property, while using the PF and MIL tracker yield camera 22. But by intuitive observation, we can see that the fused tracker always selects the best performed one. For example, at Frame_212 and Frame_214, the PF tracker is selected while at Frame_216, the MIL tracker is selected. While in case 2, because of the poor illumination, the PF tracker performs badly. This time, the SOB and MIL trackers are more preferable, as is shown by the fused tracker. We do experiments on 2 cases for our own datasets. The results are listed in Table III. The performances of using a single tracker and the fused approach are shown in Table IV

TABLE IV. CORRECT TRACKING PERCENTAGE FOR CASE 1

Trackers used / Persons	PF	SOB	MIL	Fused
P_1	89.34	90.18	88.67	95.12
P_2	88.44	92.24	90.99	97.21
P_3	90.12	87.34	85.02	93.08
P_4	87.66	82.13	84.96	94.77
P_5	88.88	92.64	91.13	97.27

TABLE V. CORRECT TRACKING PERCENTAGE FOR CASE 2

Trackers used / Persons	PF	SOB	MIL	Fused
P_1	59.36	84.55	88.66	90.14
P_2	62.93	88.67	89.32	91.28
P_3	64.58	87.96	89.76	93.67
P_4	60.33	83.21	89.11	92.22

Figure 5. Some results for camera selection in case 3. Similar to Figure 4, only the view of selected cameras are shown. In Frame_304, the PF tracker for view is selected. This information is used to resume the SOB and MIL tracker such that in Frame_305, they are almost equally good (View 1 is still selected because it has a better view for Person 1 compared with other views). As Person 1 walks, the system selects different cameras by using different individual trackers as shown in Frame_310 and Frame_314. Due to the consideration of temporal smoothness, camera hand-offs cannot take place too frequently. This explains why in Frame_365, the system still selects the MIL tracker in view 8 and the hand-off does not happen until Frame 368.

and Table V.

2) Experiments on Public Datasets

In Figure 5, we show the tracking results on some frames by using different individual trackers and the fusion of multiple trackers on the PETS2009 S2.L1 dataset. We choose 9 persons who appear most frequently to track, but in Figure 5, we only show the results for Person 1 as denoted. For the overall results for all persons, see Table VI. In Frame_304, Person 1 can only be seen in view 1, view 6 and view 7, with view 1 has the best view. In this view, the PF tracker's final score is higher than that of the other two trackers (because it has a higher region covariance) and is selected. We use this PF tracker's information to resume other trackers in other cameras with the help of the correspondence information provided by the homographies between each pair of cameras, so that the tracking error will not propagate. We can observe that, afterwards, when the person enters the FOV of camera 8, the SOB tracker for camera 8 has the best tracking quality and is thus selected at Frame_314. In Frame_365, view 5 is better than view 8, but since temporal smoothness is taken into account, view 8 is still selected to make sure that the cameras do not hand over from one to another too frequently. The hand-off takes place in Frame_368 when view 8 is much worse than view 5. Note that the correct percentages in Table VI are low for some of the persons because when doing cross camera tracking, correspondences among cameras are important. As stated previously, we do this by manually picking up corresponding points. This makes the computed homography is not very accurate. In our experiment, we choose the views for which the homography error is smaller or equal to 10 pixels. So, when the person's size becomes smaller, this homography error will cause tracking errors easily. This problem can be solved if we have better calibration data.

As we can see from Table IV to VI, individual trackers does not work ideally under all circumstances. In the experiment case 1, the colors of the persons' clothes are required to be distinct from each other. In this case, the color-based PF tracker has a better performance compared with case 2 in which the illumination condition is poor or case 3 where the colors of persons' clothes are hard to be distinguished. However, when applying the proposed approach, we can neglect the drawbacks of an individual tracker and achieve a relatively stable result as long as there is at least one individual tracker (not necessarily to be the same individual tracker) can work reasonably well in each single frame.

TABLE VI. CORRECT TRACKING PERCENTAGE FOR CASE 3

Trackers used / Persons	PF	SOB	MIL	Fused
P_1	80.15	89.23	81.22	93.02
P_2	84.69	88.96	87.34	90.02
P_3	79.69	82.64	81.33	89.96
P_4	81.11	84.12	82.56	92.26
P_5	80.01	84.56	81.57	88.91
P_6	89.63	90.19	91.32	94.10
P_7	78.09	84.98	86.24	90.14
P_8	79.64	85.96	86.12	88.63
P_9	82.88	86.77	85.13	90.79

V. CONCLUSIONS

In this paper, we proposed an approach for the score level fusion of multiple state-of-the-art trackers for camera selection and hand-off. The fusion scheme treats each individual tracker as a black box, i.e., the fusion methodology is not affected by what kinds of trackers are fused. The merit of the proposed score-level fusion is that we can take advantage of multiple trackers under any scenarios. As long as there is at least one tracker working well, the tracking result is reliable. This is different from other feature-level fusion methods for tracking, which can be treated as a new tracker and be fused under our proposed scheme. The user-supplied criteria for camera selection in this paper consider both the tracking quality and the camera hand-off quality. The derived bounding box, accounts for the spatial-temporal smoothness of the moving object.

We provided experimental results both on our own dataset and a subset of the public PETS2009 dataset. We compare different combinations of trackers and select the PF/SOB/MIL for our experiments. The experimental results show that this approach outperforms those by using any single tracker. This is also a novel idea to solve the camera selection problem. The computational burden is very low. When using 3 trackers for a person, in a 5-person case, the processing speed can be 21 fps, such that it is easy to apply the proposed approach in real-time applications. However, it is to be noticed that in different applications, different combinations of trackers may be needed. In the future, we will work on dynamic models to select trackers adaptively.

REFERENCES

[1] G.R. Bradski, "Computer Vision Face Tracking for Use in a Perceptual User Interface," *Intel Technology Journal* Q2 1998.

[2] M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking," *ICCV* 1998.

[3] H. Grabner, M. Grabner and H. Bishof, "On-line Boosting and Vision," *CVPR* 2006.

[4] H. Grabner, C. Leistner and H. Bischof, "Semi-supervised On-line Boosting for Robust Tracking," *ECCV* 2008.

[5] B. Babenko, M. Yang and S. Belongie, "Visual Tracking with Online Multiple Instance Learning," *CVPR* 2009.

[6] K. Bernardin and R. Stiefelhagen, "Audio-visual Multi-person Tracking and Identification for Smart Environments," the 15th International Conference on Multimedia, 2007.

[7] C. Conaire, N. O'Connor and A. Smeaton, "Thermo-visual Feature Fusion for Object Tracking Using Multiple Spatiogram Trackers," *Machine Vision and Applications*, Vol. 19, Sep. 2008.

[8] Y. Li and B. Bhanu, "Utility-based Camera Assignment in a Video Network: A Game Theoretic Framework," *IEEE Sensors Journal*, Vol. 11, Issue 3, pp. 676-687, 2011.

[9] O. Tuzel, F. Porikli and P. Meer, "Region Covariance: A Fast Descriptor for Detection and Classification", *ECCV* 2006.