Color-Based Multiple Agent Tracking for Wireless Image Sensor Networks

Emre Oto, Frances Lau, and Hamid Aghajan

Wireless Sensor Networks Lab
Department of Electrical Engineering
Stanford University, Stanford, CA 94305
{eoto, flau, aghajan}@Stanford.edu

Abstract. This paper presents an implementation of a color-based multiple agent tracking algorithm targeted for wireless image sensor networks. The proposed technique is based on employing lightweight algorithms and low-bandwidth data communication between multiple network nodes to track the path of autonomous agents moving across the fields of view (FOV) of the sensors. Segmentation techniques are applied to find the agents within the FOV, and a color histogram is constructed using the hue values of the pixels corresponding to agents. This histogram is used as a means of identification within the network. As such, the algorithm is able to reliably track multi-colored agents of irregular shapes and sizes and can resolve identities after collisions. The proposed algorithm has low computational requirements and its complexity scales linearly with the size of the network, so it is feasible in low-power, large-scale wireless sensor networks.

1 Introduction

The problem of tracking people, cars and other moving objects has long been at the focal point of many technical disciplines. Many techniques have been proposed and implemented to track multiple objects with multiple cameras, most of which employ stochastic models such as Kalman filtering, particle filtering, and condensation algorithms to overcome problems of occlusion, noisy observations and other visual artifacts [1]. Nguyen et al. [2] have implemented a distributed tracking system employing a Kalman filter to track multiple people within a room monitored by multiple cameras with overlapping fields of view. A similar solution to the same problem has been proposed by Chang et al. [3], who use a Bayesian network and Kalman filtering to establish correspondence of subjects between subsequent frames. There have also been applications of multiple object tracking algorithms implemented in sensor networks, such as the method demonstrated by Chang and Huang [4], in which distributed processing between multiple trackers is employed and data fusion is realized by an enhanced Kalman filter.

Color-based tracking methods that combine color information with statistical methods have also been in practice. Liu et al. [5] have implemented an algorithm based on particle filtering and color histogram information for object tracking. A

J. Blanc-Talon et al. (Eds.): ACIVS 2006, LNCS 4179, pp. 299 – 310, 2006.

similar color-based Kalman particle filter algorithm has been implemented for the same application by Limin [6]. Perez et al. [7] have used a hue-saturation histogram with a particle filter based probabilistic technique for tracking in cluttered environments. One common aspect of all the mentioned color-based algorithms is that they have been designed for use outside of the wireless sensor networks domain, and require significant processing load by a centralized processing unit.

In this paper, we propose a hue histogram based multiple object, multiple camera tracking algorithm that is intended to be simple in nature to find potential use in wireless sensor nodes. The main contribution of this paper is to demonstrate that identity management of multi-colored agents of irregular shapes traveling on a cluttered background in an image sensor network can be performed using deterministic histogram matching techniques on the hue histogram, a metric that is small in size to be communicated between sensor nodes, and yet provides reasonable resilience against changes in illumination.

The Color-Based Multiple Agent Tracking (COBMAT) algorithm uses multiple image sensors to track the movement of autonomous mobile agents or targets traveling within overlapping or non-overlapping fields of view (FOVs) of overhead and side view cameras. The objective of the algorithm is to keep track of the agent positions in a distributed fashion, without the need for a centralized control scheme. The algorithm achieves this by communicating agent information between sensors and by performing identity management using hue histograms. The algorithm does not predict the path of agents traveling between non-overlapping FOVs. This is coherent with practical applications where cameras may be used to monitor specific areas of a field, e.g. rooms of a building or particular sectors of a military field.

The image sensors are assumed to be localized by an algorithm such as in [8]. In some tracking applications, the objective may be to know which network node is tracking the agents or targets of interest. In other applications, the global coordinates of the agent at observation times may also be required. In the case of overhead cameras, the agent's global position in the area spanned by the network can be calculated by the tracking node and broadcasted to other nodes. In the case of side view cameras, exchange of image plane agent positions between the nodes that simultaneously observe the agent can result in determining the global position of the agent.

The COBMAT algorithm relies on a background subtraction and segmentation routine to obtain the blobs in each frame. The color histogram of each blob is extracted by calculating the hue of each pixel from the RGB (red, green, blue) values and then binning the hue values to create a histogram. This hue histogram is compared to those belonging to the agents identified in previous observations to associate new blobs with agents tracked in the previous FOV. The blobs that could not be identified as existing agents are compared against a local database called the "Potential Agents Database", which contains the histograms of the agents detected by neighboring sensors. Each node broadcasts every new agent entering its FOV to its one-hop neighbors and each sensor records these messages in its "Potential Agents Database". This is done to determine if the new agent has been previously identified by the network or if it is a new entry to the realm of the network.

2 System Overview

COBMAT is an algorithm intended for use with image sensor nodes in tracking agents or targets within the field of view (FOV) of one sensor and in providing inter-sensor broadcast of information on agents or targets traveling between the FOVs of different sensors. Fig. 1 illustrates a schematic for a network of image sensors covering an area of monitoring interest.

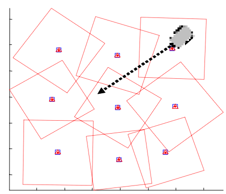


Fig. 1. Illustration of the network of overhead image sensors and a moving agent

The overall operation of the COBMAT algorithm can be epitomized as in Fig. 2. Within a single FOV, the blobs are extracted with the Blob Extraction Module, and then identified using the Identity Management Module. The Inter-sensor Communication Module handles the broadcast of new entries to a sensor's FOV to its one-hop neighbors. These three modules are described in the following sections. The distributed nature of COBMAT enables it to be scalable to large camera networks.

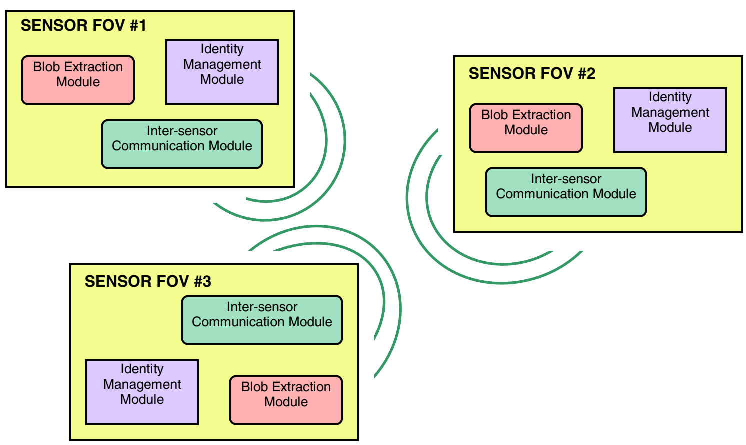


Fig. 2. Overall system block diagram of the COBMAT algorithm

2.1 Blob Extraction

Blob extraction is performed on the still images obtained at each frame instant. This operation relies on background subtraction followed by a segmentation routine. Background subtraction identifies the change mask by thresholding the difference between the intensities of each of the color channels in the current frame and the background image. To be identified as a changed pixel, the pixel values in any of the red, green or blue channels of the pixel must have changed by more than a certain threshold. During experimental studies, this threshold was selected according to the illumination of the environment. In a practical system, however, the threshold must be adaptively determined as a function of the ambient illumination.

Due to the manner in which changed pixels are found, it is important that the background image manifest the current lighting conditions. Thus, background subtraction assumes that an up-to-date image of the sensor FOV devoid of agents is available. A measure that has been taken against illumination changes is color balancing of the background image and the frame prior to background subtraction. A color balancing algorithm based on the gray-world assumption has been observed to eliminate spurious blobs arising in the case of low illumination.

Following background subtraction, a small-region removal is run on the mask image to remove artifacts due to slight noise effects. This is followed by a large-region removal algorithm, applied on the inverted mask, to fill the holes within the regions identified as blobs. These regions are then labeled, and the mask resulting from the two region removal operations is applied on the original image to retrieve the blob pixel values. The hue of each pixel is then calculated and the hue histogram of each blob is produced. The hue histograms used in this study were of 36 bins.

The position of the blob, i.e. the target location within the FOV, is found as the median of the rows and the median of the columns comprising the blob.

2.2 Identity Management

In each frame instant, the hue histograms of the blobs are extracted, and these histograms are compared to those belonging to the agents in the previous frame to identify which agents have remained in the FOV, which agents have just entered, and which have left the FOV. The histogram matching routine employed at this stage relies on two different distance measures to decide whether the two histograms are the same: the Euclidean Distance (ED) and the Vector Cosine Distance (VCD) measures. The ED between two histograms h[n] and g[n] of length N is given as in (1).

$$\mathrm{ED} = \sqrt{\sum_{n=1}^{N} (g[n] - h[n])^2}\ . \tag{1}$$

Treating the histograms h[n] and g[n] as two vectors in R^N, ED is the norm of the difference vector h[n]-g[n]. VCD is a measure of proximity proposed by Sural et al. [9] that derives itself from the Euclidean geometry, and is the angle between the two vectors. This projection angle θ(g[n], h[n]) can be calculated as given below:

$$\theta(g[n], h[n]) = \cos^{-1}\left(\frac{\sqrt{\sum_{n=1}^{N} (g[n] \cdot h[n])}}{|g| \cdot |h|} \right), \tag{2}$$

where $\left|g\right| = \sqrt{\sum_{n=1}^{N}(g[n])^2}$ and $\left|h\right| = \sqrt{\sum_{n=1}^{N}(h[n])^2}$.

Two agent histograms are inferred to be the same if both the ED and VCD tests yield that the histograms are the same, which is decided if the distance calculated by the two measures are individually smaller than the thresholds. Through experimentation, an ED of 1 and a separation angle of $\pi/4$ were found to be appropriate thresholds for correct detection even under varying illumination conditions.

It can easily be observed that both the ED and VCD are scalar norms calculated on a bin-by-bin basis. By their nature these norms fail even if the histograms are very similar in shape but are shifted with respect to one another. The effect of such a shift can be amplified in the case of the hue histogram since it is an angular measure wrapping around from 2π to 0 radians. To alleviate the bin shift and wrap around problems, the ED and VCD are calculated as follows:

$$ED_{g,h}= \min(ED(g[n],h[n]), ED(g[n],h[((n-1))_N]), ED(g[n],h[((n+1))_N]), ED(avg(g[n]),avg(h[n]))) \quad (3)$$

$$\theta_{g,h}= \min(\theta (g[n],h[n]), \theta (g[n],h[((n-1))_N]), \theta (g[n],h[((n+1))_N]), \theta (avg(g[n]),avg(h[n]))) , \quad (4)$$

where $((\ ))_N$ represents a circular shift in modulo N, and the avg() function is the application of an averaging window of size N_w. The circular shift by 1 covers single bin shifts of the histograms with respect to each other, while the window averaging leverages the result by smoothing when the two histograms are shifted by more than a single bin. In our empirical studies, we observed that $N_w = 3$ achieved sufficient smoothing.

Fig. 3 depicts the identity management algorithm employed in each sensor. The histograms of agents traveling in the FOV in the most recent frame are stored in cache memory for easy access when performing identity associations of agents in the current frame. At each frame instant identity association is performed by means of histogram matching as explained above. If a blob in the current frame can be matched to any of the agents, the position of that agent is updated in the cache.

If no match can be found in the current FOV, then the position of the blob is considered. If all the cameras are overhead cameras, then the background on which the agents travel is a 2-D plane, so new agents may not emerge from inside the FOV. Using this fact, we only consider the blobs that emerge near the edge of the FOV to be new agents, and assume all other blobs are artifacts caused by sudden lighting changes. The threshold distance used to decide if the agent is far away from the edge or not was determined by experimentation as it depends on the size of the FOVs and the agents as well as the frame rate and the range of possible agent speeds. In the situation where side-view cameras are used, this condition for the position of the blob is not used.

Agents that are new to a FOV are first sought in the "Potential Agents Database", which contains the histograms and labels of the agents communicated by the sensor's one-hop neighbors upon entrance to their FOVs. If the agent is identified in the"Potential Agents Database", the matching entry is copied from the database to the cache. If the

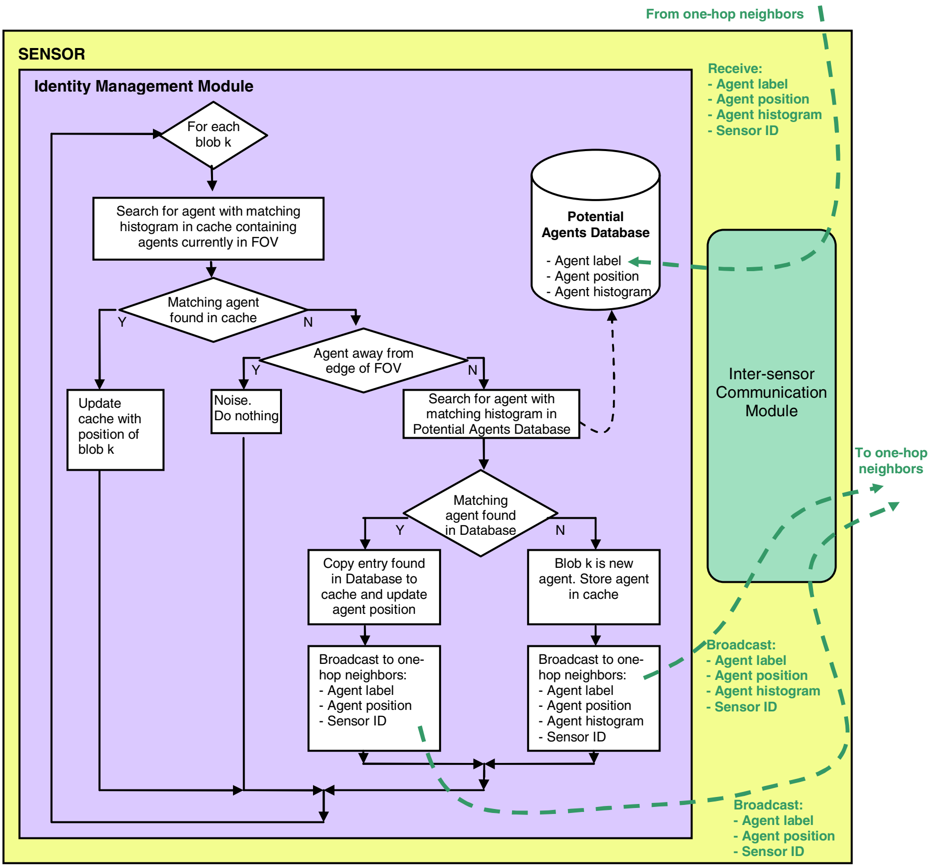


Fig. 3. Identity management algorithm

agent cannot be matched to any of those in the database, then it is declared as a new entry to the network, and the agent is stored in the cache and a copy is also stored in the database. The node then broadcasts a message to its one-hop neighbors, as described in the next section.

2.3 Inter-sensor Communication

Once a new agent is registered to the cache, the corresponding agent label is updated with the current timestamp, and this label along with the current position and the sensor ID is communicated to the one-hop neighbors. If the agent is also a new entry to the network, then the agent histogram is also transmitted.

This dissemination algorithm informs all neighboring nodes about the entry of an agent into the FOV of the sensor. Each sensor also listens to transmissions from its one-hop neighbors and keeps its "Potential Agents Database" up-to-date. Currently, the algorithm has been designed such that the sensor constantly listens for incoming packets.

A networked imaging system as devised here allows any end user at a sink node to make queries about some or all of the agents currently being tracked by the network. Specifically the user may wish to know the positions of the agents being tracked relative to a global coordinate system on the area encompassing the sensor network. Agent position information can be included in the packet sent to one-hop neighbors to allow for data aggregation through gateway nodes in a large sensor network. A query-based protocol can then collect the positions and identities of agents traveling in different FOVs for a centralized observer node.

To report agent positions, the coordinates of an agent relative to the FOV must be transformed into the global coordinate system. For an overhead camera system, the nodes could be localized using a method such as in Lee et al. [8], and the following transformation can be used to convert coordinates within a sensor's FOV to the global coordinates:

$$s_i = (\alpha\, R)^{-1}\, y_i + p\ . \qquad (5)$$

In this equation, y_i are the coordinates of the agent in the FOV in which it is traveling, s_i are the coordinates in the global coordinate system, p is the vector extending from the origin of the global coordinate system to the origin of the sensor's FOV, α is the number of pixels in the FOV corresponding to one inch on the ground, and R is the rotation matrix with theta as the rotation in radians.

3 Results and Discussion

The networked tracking scheme described above operates under the assumption that the hue histograms of agents are invariant of the cameras and camera placements so that objects can be matched between different FOVs possibly observing different lighting conditions. This assumption was verified using a four-camera setup to compare the hue histogram produced by the different cameras. The effect of illumination on the hue histogram was investigated with a single camera and different illumination conditions. These experiments and their results are presented in the following sections.

3.1 Tracking Within the FOV

Fig. 4 illustrates an example of tracking two multi-colored agents traveling within the FOV of a camera. The displayed hue histograms are used to track the position of the agents as they appear in the subsequent frames captured by the camera.

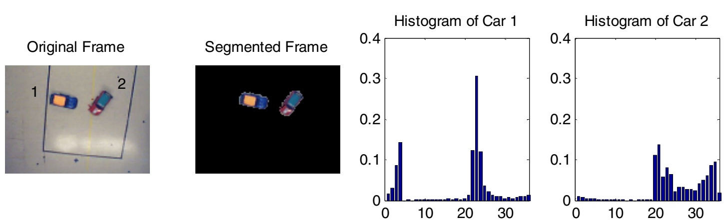


Fig. 4. Tracking multi-colored agents within the FOV of a camera using their hue histograms

3.2 Multiple FOV Operation

Three overhead cameras of the same model and one oblique camera of a different model were placed facing the floor of our lab, and three remote controlled cars were driven through their FOVs. The results of the experiment are as in Fig. 5.

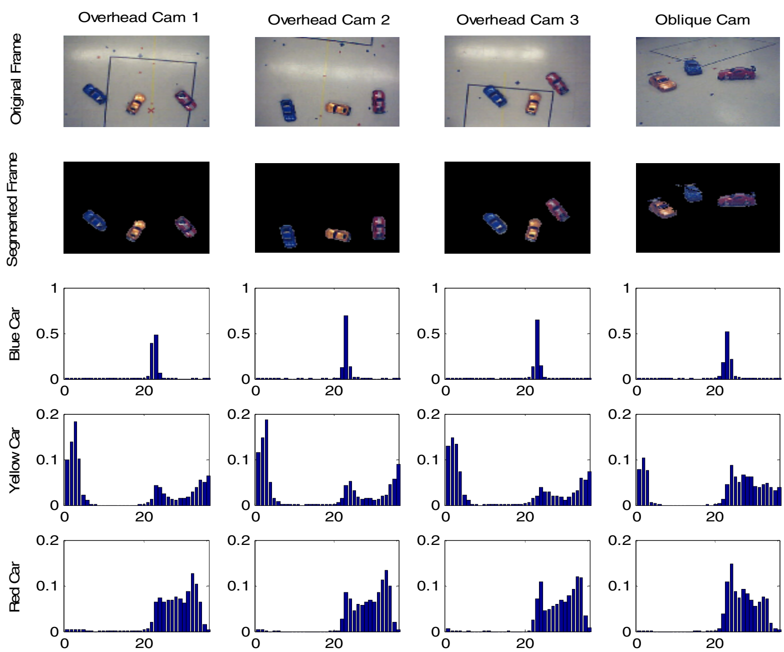


Fig. 5. Comparison of hue hisograms produced by four different cameras for three agents. Each bin on the x-axis corresponds to ten degrees of hue.

The first, second, and third cars were blue, red, and yellow, respectively, but each car had details in other colors. It was observed that within a single FOV, both the overhead cameras as well as the oblique camera tracked the objects reliably. It is worth noting that the oblique camera tracked the three cars successfully regardless of perspective effects. However, the oblique camera could only track agents successfully under the assumption that the agents did not occlude each other. Occlusions could be resolved by the collaboration of two or more cameras, which has not been addressed in this work. Frames were captured from the four cameras and hue histograms were produced for each of the segmented blobs as presented in Fig. 5. The histograms obtained through this experiment demonstrate that each car's hue histogram retained its form between the cameras, despite the fact that the fourth camera had an oblique view and was of a different model.

3.3 Variations in Illumination Condition

The variation of the hue histogram within a single FOV with different lighting conditions was investigated for the same three agents presented in Sec. 3.2, and it was observed that for all the three agents the histogram remained almost the same with varying illumination level. Fig. 6 presents the hue and intensity histograms for the yellow agent under three different illumination settings.

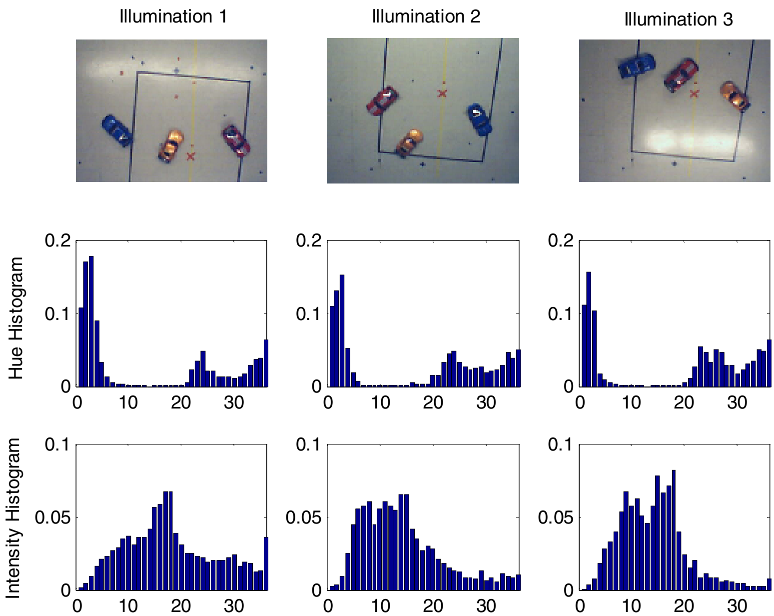


Fig. 6. Comparison of hue and intensity histograms for the yellow agent produced in three different illumination conditions. Each bin on the x-axis corresponds to ten degrees of hue.

The intensity histograms change depending on the illumination level. Therefore, by only considering the hue, the algorithm demonstrates robustness against changes in illumination. We did not have precise control over the illumination of the room and could not investigate the effect of larger changes in illumination. However, it was observed that the algorithm fails under very low illumination, such as when all the lights in the room are turned off and there is only a small amount of light entering through the door. It also fails under very heterogeneous illumination, that is, when part of the FOV has extremely low or extremely bright lighting. This result is expected since it is known from [9] that the hue histogram is only a definitive measure when the saturation is relatively high and the intensity is not extremely low or high. When the saturation is very low, the color becomes very pale and it is difficult to detect the hue. When the intensity is extremely low or high, the image is so dark or bright, respectively, that the hue is not apparent. In these cases, a color is better represented by its grayscale intensity rather than its hue. A possible solution to this problem is proposed in Sec. 4.

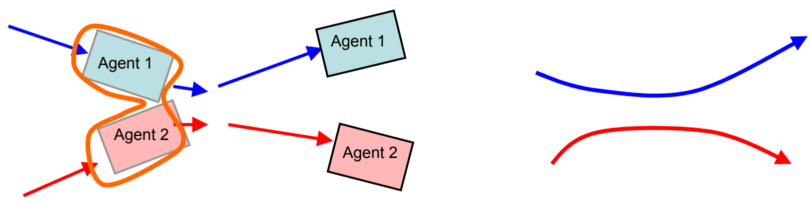
3.4 Collision Handling

COBMAT differentiates between agents after a collision with the same method as when no collisions occur, namely, by matching the hue histograms. Therefore, the elegance of our solution is that we use one technique to handle both tracking without collisions and tracking with collisions. This ability to handle collisions is the primary advantage of attribute-based tracking versus path history-based tracking.

The current implementation of COBMAT cannot track multi-colored agents during a collision but can recover the agent identities right after the collision. If the collision occurs at the edge of the FOV, the sensor assumes that the merged blob is a new agent and broadcasts it to neighboring nodes, causing a false alarm. That condition needs special handling via predicting a collision when two agents approach each other in an edge zone, and is the subject of further investigation. However, by constraining the agents to single-colored objects, we were able to reliably track the agents even during collisions. Figure 7(a) shows the situation that occurs when agents collide: The blobs merge. Figure 7(b) illustrates how a tracking scheme would resolve the collision by taking the following steps to localize the agents within the merged blob.

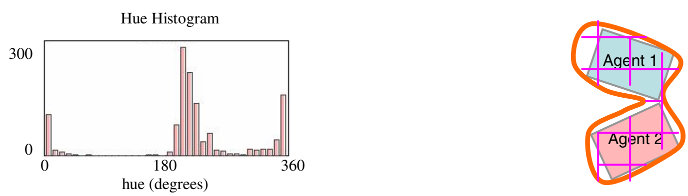
1) Create a histogram for the merged blob.
2) Determine the number of peaks. Since the agents are single-colored, the number of peaks equals the number of agents in the merged blob (see Fig. 8(a)). Record the hue value of these peaks.
3) Divide the merged blob into sub-blocks (see Fig. 8(b)).
4) Create one histogram for each sub-block.
5) For each hue value recorded in step 2, determine which sub-block has a histogram with the greatest amplitude at this hue. The center of this sub-block is reported as the center of the agent with that hue. This is a reasonable estimate since the sub-block with the largest amplitude at a certain hue has the largest area that is of that hue.

An experimental test result for the case of single-color agents is depicted in Fig. 9.



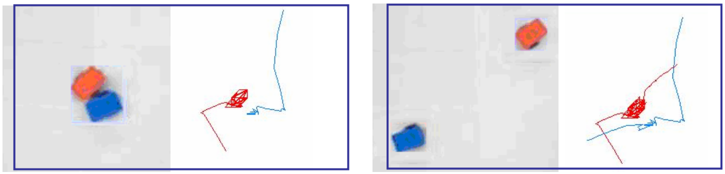
(a) (b)

Fig. 7. (a) Merging of blobs in a collision (b) Resolved paths during a collision



(a) (b)

Fig. 8. (a) Two peaks in histogram indicate two single-color agents merged (peaks at 0 and 360 degrees both correspond to the red agent). (b) Scheme for localizing agents within merged blob.



(a) At the collision (b) After the collision

Fig. 9. Experimental result of the collision handling scheme for single-color agents, showing an illustration of the path plotted with the coordinates obtained

3.5 Computational Complexity

Since this algorithm is targeted for wireless sensor networks with low-power and low-cost sensor nodes, low computational complexity is one of the main objectives.

This algorithm achieves this goal by employing simple and elegant techniques designedwith the specific needs of wireless sensor networks in mind. The computational complexity C for each sensor to analyze one frame is approximately

$$C = \alpha \cdot numAgents \cdot (\beta \cdot agentSize + \gamma \cdot numBins) + \kappa \cdot imageSize, \tag{6}$$

where *numAgents* is the number of agents in the FOV, *agentSize* is the size of each agent (in pixels), *numBins* is the number of bins used in the histograms, *imageSize* is the size of the FOV (in pixels), and α, β, γ, and κ are constants. κ accounts for the processing required to extract the blobs and γ represents the histogram matching computations. This algorithm is remarkably efficient because it is only linearly proportional to its parameters. Due to the broadcasting nature of transmissions, the network's operational complexity is also linear in the number of nodes. This makes the algorithm scalable, and hence feasible in large-scale wireless sensor networks.

4 Conclusions

A lightweight technique based on color histograms has been proposed for tracking multiple agents in a distributed image sensor network. The algorithm employs simple image processing and limited data communication between the nodes. The algorithm operates by exploiting the invariance of the hue histogram in different camera and different illumination settings to differentiate and match agents traveling within the network. It has been demonstrated that the hue histogram can be used reliably when the illumination within the FOV does not show drastic changes on the background. To adapt the current scheme to backgrounds with large illumination gradients, e.g. the floor of a forest, the histogram matching algorithm must be made adaptive to consider larger shifts of the hue histogram depending on the variation in the corresponding intensity histograms. It is also known that the hue histogram is not a reliable metric under very low illumination. To alleviate this effect, it may be possible to use hue and intensity histograms interchangeably depending on the saturation level of the image, stemming from the ideas presented in [9] and discussed in Sec. 3.3. However, the additional use of the intensity histogram will double the data overhead incurred.

References

1. Marcenaro, L., Ferrari, M., Marchesotti, L., Regazzoni, C.S., Multiple object tracking under heavy occlusions by using kalman filters based on shape matching. Int. Conf. on Image Processing, Rochester, NY (2002) 341–344
2. Nguyen, N., Bui, H.H., Venkatesh, S., West, G., Multiple camera coordination in a surveillance system. ACTA Automatica Sinica, Vol. 29 (2003) 408-422
3. Chang, T.H., Gong, S., Tracking Multiple People with a Multi-Camera System. IEEE Workshop on Multi-Object Tracking (2001) 19–26
4. Chang, C.K., Huang, J., Video surveillance for hazardous conditions using sensor networks. IEEE Int. Conf. on Networking, Sensing & Control, Taipei, Taiwan (2004) 1008–1013
5. Liu, F., Liu, Q., Lu, H., Robust Color-Based Tracking. 3rd Int. Conf. on Image and Graphics (2004) 132–135
6. Limin, X., Object tracking using color-based kalman particle filters. IEEE Int. Conf. on Signal Processing, Beijing, China (2004) 679–682
7. Pérez, P., Hue, C., Vermaak, J., and Gangnet, M., Color-Based Probabilistic Tracking. 7th European Conf. on Computer Vision-Part I (2002) 661–674
8. Lee, H., Aghajan, H., Vision-Enabled Node Localization in Wireless Sensor Networks. COGnitive systems with Interactive Sensors, Paris, France (2006)
9. Sural, S., Qian, G., Pramanik, S., A histogram with perceptually smooth color transition for image retrieval. 4th Int. Conf. on Computer Vision, Pattern Recognition and Image Processing, Durham (2002)