A Multi-Agent Framework for Visitor Tracking in Open Cultural Places

Muhammed Safarini[1], Rasha Safarini[1], Thaer Thaher[1], Amjad Rattrout[1], and Muath Sabha[1,*]

Arab American University, Jenin, Palestine,

Abstract. Real-time tracking of pedestrians has been attracted tremendous attention for many purposes, especially studying their behavior. The perfect tracking is still a challenging open research. In this paper, a novel approach was introduced by employing a multi-agent vision-based system to achieve an accurate real-time tracker. The proposed approach is recommended to be used in open cultural places where there may be some obstacles including buildings, columns, and many others. The proposed model efficiency was assessed on four real Multiple Object Tracking (MOT) challenge benchmarks in terms of Multi-Object Tracking Accuracy (MOTA). Simulation results reveal high efficacy of the proposed model compared with the state-of-the-art techniques.

Keywords: Multi-agent system, multi-person tracking, JADE, cultural heritage, computer vision.

1 Introduction

Currently, the detection and tracking of pedestrians in public places is attracting considerable attention from the research community. It can be exploited for several purposes such as surveillance and security-related issues [1]. In public and open spaces, the targeted visitors to be tracked are not expected to be provided with any transmitter or marker for tracking purposes. Consequently, Computer stereo vision plays an important role in tracking multiple visitors through the analysis of real-time video streams from multiple fixed surveillance cameras [2] [3].

This process is often carried out in two stages: detecting and tracking. Human detection, which is a branch of object detection, is utilized to identify the presence of objects and localize humans (i.e., identify the location of humans in rectangular bounding boxes) [4]. Various machine learning approaches have been introduced for human detection such as Haar cascade [5], Histograms of Oriented Gradient (HOG) [6], and deep learning-based methods [7]. However, these techniques require high computation power. In this paper, an efficient framework for multi-person tracking in open cultural places was proposed by combining Multi-Agent System (MAS) and computer vision methods. The main contributions are:

- Finding a framework that tracks humans by putting an agent for each.
- Finding a technique which could achieve real time tracking on parallel computing.
- Finally, achieving a successful tracking framework using MAS.

The rest of paper is organized as follows: The related works are reviewed in Section 2 and Section 3. In Section 4, the proposed methodology involved in this work is presented and discussed deeply. Section 5 presents the simulation results and their analysis. Finally, Section 6 concludes the overall results and future works.

2 Review of related works

The perfect real-time tracking is still a challenging problem. Several aspects make this process very difficult, especially for humans. Their clothes vary in colors and texture, which increases the complexity of tracking. Besides the random behavior of humans and interactions. For instance, people may move singly for a while or move together as a group for another. Moreover, many challenges arise due to several factors, including varying lighting, motion blur, crowded scenes, partial occlusions (the human is not entirely observed), cluttered background, and other environmental constraints. For tracking purposes, machine learning classifiers are employed to learn from the detected objects in the previous and subsequent frames; thus, additional computational power is required [4].

Due to the challenges mentioned above, the tracking by detection problem is still open for research. The performance of such systems is based on two contradictory metrics; the maximum accuracy of the final results and the minimum computation time [3]. Recently, Multi-Agent Systems (MASs) have attracted considerable attention from researchers in the field of computer vision. It is highly recommended to deal with complex problems effectively [8].

Srisamosorn et al. introduced an indoor human tracking system by involving the moving robot (quadrotor) and multiple fixed cameras. The fixed cameras are used to identify the position and direction of each human, as well as the position of quadrotor. While the quadrotor is utilized to follow the human's movement by tracking his/her face based on the attached camera [9]. Although satisfactory results were achieved, the system is applicable to a small closed environment with a limited number of persons.

Choi et al. [10] introduced an efficient real-time multi-person tracking framework for Intelligent Video Surveillance. The purpose is to detect and track humans without supervision. Their methodology uses the background subtraction approach to extract the Region of Interest (ROI) and utilizes particle filtering for tracking purposes. Sabha and Nasra [11] used fuzzy logic to give accurate results and reduce the ambiguity on user recognition. In addition to that, they reduced the calculation by using parallel computation.

Jafari et al. [12] presented a real-time multi-person tracking engine based on Red, Green, Blue, and Depth (RGB-D) vision. Their proposed system is suitable for mobile robots and head-worn cameras. The authors speed up the detection process by exploiting the available depth information from the current RGB-D sensors.

A system for real-time closed-space tracking of persons in a shopping mall was proposed by Bouma et al. [13]. The system was tested with multiple static cameras without overlapping field-of-views. The system consists of three main components called: Tracklet generation, Re-identification engine, and graphical man-machine interface. Sabha and Abu Daoud suggested a system for adaptive camera placement in open cultural places using Voronoi Diagrams [14]. The system applied a technique for the compensation of missing drones by other existing drone cameras. Multiple cameras produce one image.

Recently, MAS has been utilized in real-time multi-persons tracking tasks and showed successful performance results. However, to the best of our knowledge, there is a lack of exploiting MAS for tracking and prediction of visitor movement in open cultural places. Previtali and Iocchi [15] presented a method for Distributed Multi-Agent Multi-Object indoor Tracking using a Multi-Clustered Particle Filtering technique. They used a team of mobile sensors (robots) that keeps track of several moving objects.

Sanchez et al. [3] introduced a computational model for tracking and classifying moving objects (pedestrians and vehicles) through surveillance cameras. Their methodology is based on the integration between an intelligent MAS and information fusion techniques with the aim of minimizing the computational effort and producing a reliable accuracy. An efficient multi-object tracking approach based on Multi-Agent Deep Reinforcement Learning (MADRL) was proposed by Jiang et al. [16]. They adopted YOLO V3 to detect the objects which represent the agents.

3 Multi-Agent System (MAS)

MAS or self-organized system has been recently emerging as a promising subfield of Distributed Artificial Intelligence (DAI) [8]. It is a computerized system compound of multiple smart and autonomous entities, called agents which can collaborate with other entities, and perceive its environment to perform the desired actions [17]. MAS is based on the idea of a cooperative work environment in order to handle complex adaptive problems that are impossible to handle by the traditional centralized approaches [8]. In MAS, the whole task is divided into multiple smaller tasks allocated to the collaborative agents. To deal with complex decision-making systems and to achieve the desired goals efficiently, the agent (which can be software, hardware, or a hybrid of both) enjoys the Autonomy, Sociability and Pro-activity properties [8].

MAS has received tremendous attention from the research community in various disciplines, including graphical applications, civil engineering, smart grids, networking, manufacturing, and recently in tracking multiple moving objects [3]. In this work, a hybrid MAS model of reactive and proactive agents is exploited to tackle the problem of tracking multiple pedestrians in open cultural places.

4 The Proposed System

4.1 Overview

Existing real-time tracking approaches based on computer vision are still facing some problems, namely; the high computational demand of computer vision, the limited field of view of the used sensors, and the possibility of pedestrians tracking loss. On the other hand, in some approaches, MASs are utilized to track moving objects. However, they were applied to indoor environments.

In this paper, a novel approach is proposed by utilizing a multi-agent system combined with computer vision in order to achieve real-time tracking of pedestrians, as in Figure 1.

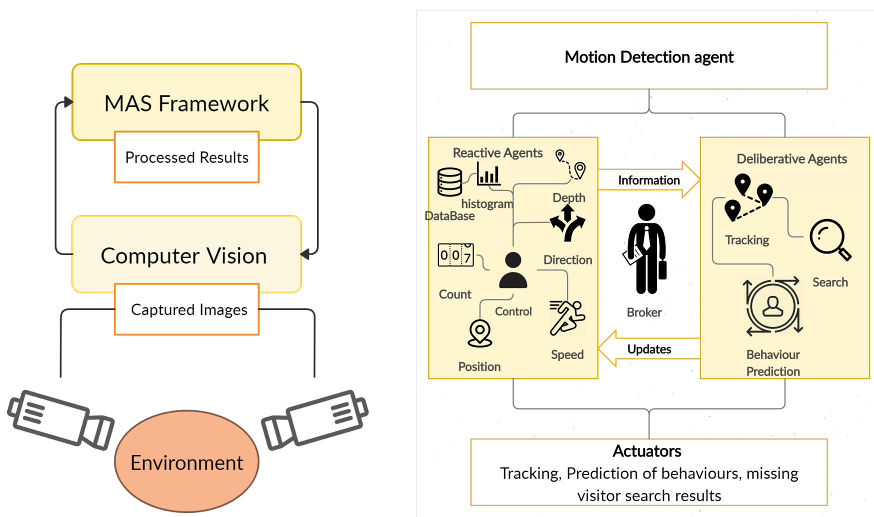


Fig. 1: The right figure represents the proposed hybrid MAS model for the tracking pedestrians framework shown in the left figure

The proposed system is designed to be applied to open historical environments with several entrances and exits. The site has to be equipped with a fixed camera. Normally, visitors of all ages and cultures enter and tour the place. The environment has the following characteristics:

- The number of visitors change dynamically over time.
- Visitors can move in groups or individuals.
- There are a set of rules that regulates cultural places pedestrians have to obey.

The developed system consists of several agents that cooperate with each other to track all visitors who enter the historical place. The system consists of eleven agents, namely; visitor, motion detection, histogram, control, position, tracking, search, speed, count, direction, static historical objects, and behaviour prediction agents.

4.2 MAS Agents

A hybrid model is used in the proposed MAS. The hybrid model is a combination of two subsystems, namely; reactive and deliberative. The reactive subsystem is able to react fast to events without complex reasoning, and the deliberative subsystems draw plans and make decisions using symbolic reasoning. The agents' interactions and the proposed model are shown in Figure 1, while Table 1 presents a detailed description of the employed agents.

Table 1: Description of employed agents

Agent Name	*Perception*	*Action*	*Goal*
Motion detection agent	Moving objects in the environment	1- Stores captured images from the camera. 2- Sends images to control agent	Detects when visitors arrive to the historical place.
Control agent	Images from motion detection agent	1- Analyzes images received from motion detection agent 2- Sends the detected objects to the histogram agent 3- Sends the position to the Position agent	1- Detects moving objects 2- Interacts with other agents
Histogram agent	Objects	1- Receives objects from control agent 2- Calculates histogram for each object 3- Compares with the stored histograms in the database.	1- Calculates histograms 2- Stores visitor histogram in the database
Position agent	position of visitor agent	1- Receives the position of a visitor, calculates the direction and stores it. 2- Sends the position of a visitor agent to either control, search, or behavior prediction agents.	Stores position of a visitor agent
Direction agent	Direction of movement of visitor agent	1- Receives and stores the movement direction of the visitor agent. 2- Sends the Direction of the visitor to either search agent or behavior prediction agent	Stores Direction of movements of each visitor agents
Depth agent	Distance between visitor agent and camera	Receives the distance between the visitor and the camera	Stores the distance between the visitor and the camera
Speed agent	Speed of visitor agent	1- Receives the speed of the visitor agent 2- Calculates and Stores the speed of each visitor 3- Sends an alarm if the speed is exceeded.	Stores the speed of the visitor agents
Count agent	Message of visitor arrival or leave	1- Receives message from control agent for new arrivals or departures. 2- Sends the number of visitors when needed	Stores the number of visitors in the place at any given time
Tracking agent	Movement of visitor agent	1- Receives position of visitor agent Continuously 2- Calculates and stores the visitor path 3- Send the path when needed	Stores the path of the visitor agent's movement
Search agent	Manages lost visitors	1- Receives information about the lost agent from other agents (position, direction, etc.) 2- Retrieves historical data from the database 3- Predicts where the lost agent could be	Search for lost visitor agent
Behavior Prediction agent	All information about the visitor agent	1- Receives information from other agents (position, direction, path, etc.) 2- Predicts the next move of the visitor agent.	Predicts the next move of the visitor agent

4.3 Methodology

The developed system consists of several agents that cooperate with each other to track all visitors who enter the historical place.

When a visitor agent arrives, the control agent detects the moving objects in images from the cameras using Open Source Computer Vision (OpenCV) contour function and sends the moving objects to the histogram agent. Then the histogram agent calculates the histogram of each object and compares it with the stored histograms in the database using a specific threshold to decide if the person in this frame is already registered in the site or he is a new visitor. If

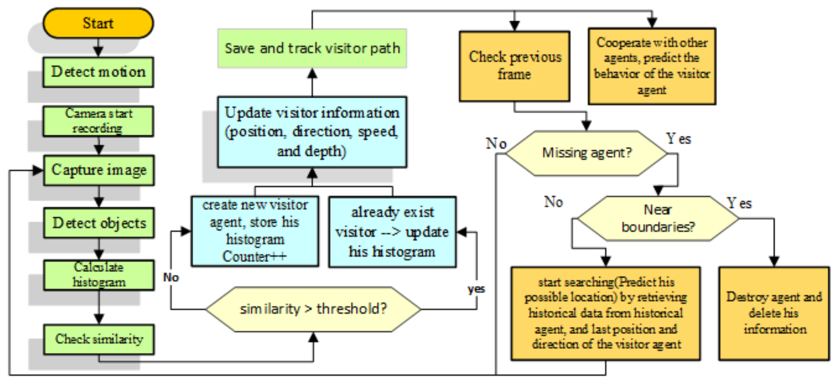


Fig. 2: System methodology

the detected person is already there, it updates his histogram. If the histogram does not match any of the stored histograms, this means he is a new visitor. The histogram agent stores the new histogram in the Database (a new visitor is registered), and sends a message to the count agent to increase the number of visitors by one.

To express the matching between two histograms H_1, H_2 we employed the OpenCV correlation metric [18] calculated as in Eq. (1):

$$d(H_1, H_2) = \frac{\sum_I((H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I(H_1(I) - \bar{H}_1)^2 \sum_I(H_2(I) - \bar{H}_2)^2}} \tag{1}$$

$$where\ \bar{H}_k = \frac{1}{N}\sum_J H_k(J)\ and\ N\ is\ the\ total\ number\ of\ histogram\ bins.$$

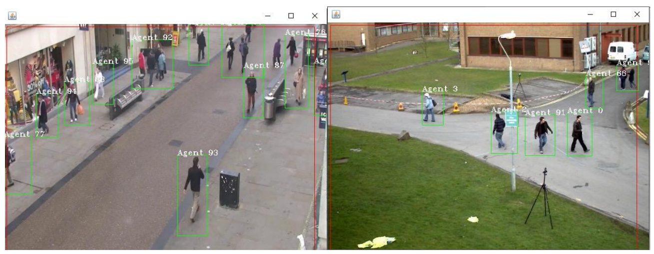
The control agent also interacts with depth, speed, position, direction, and tracking agents to store the needed information about the visitor. With each movement of the visitor agent, the motion detection agent keeps sending the captured images to the control agent, which in turn interacts with other agents to update his information to keep him tracked and store his movement path. If the visitor agent goes out of the covered area, then the control agent informs other agents to delete his stored data and decreases the counter by one.

The search agent works when a visitor is lost within the environment limits, the agent gets the historical statistical data, and its last (x, y) position of the lost visitor to predict his possible location. Finally, the behavior prediction agent learns the visitors' behavior using his location history and predicts the next place the visitor could go to. The whole system is described in detail in Figure 2.

5 Simulation Results

Simulation Setup To assess the tracking performance of the proposed algorithm, four well know MOT challenge benchmarks were employed in this research [19]. These benchmarks were chosen carefully with different details and complexity to cover various behaviors. The prototype of the framework has been designed and implemented using Java Agent Development (JADE) [20] platform and OpenCV. The used Java development environment is a PC with Intel Core i5-4200, 2.3 GHz CPU, 16 GB RAM.

The algorithm was applied on four real MOT open access videos which are PETS09-S2L1, AVG-TownCentre, TUD-Crossing, and Venice-1 shown in Figure 3. The tracking performance of the proposed system was assessed and compared with other existing approaches based on Multi-Object Tracking Accuracy (MOTA) measure. It is a commonly used evaluation metric which indicates how many mistakes the tracker made in terms of missed targets, false positives, mismatches, and failures to recover tracks [21]



(a) AVG-TownCentre (b) PETS09-S2L1

(c) TUD-Crossing (d) Venice-1

Fig. 3: Videos where the algorithm is applied with bounding boxes around the detected pedestrians.

The Impact of Similarity Threshold The threshold is the desired lower limit for the similarity of two detected objects that belong to different frames. It is a vital factor that needs to be properly tuned. Extensive experiments using different values of threshold (0.65, 0.7, 0.75, 0.8, and 0.9) were conducted for each tested video. From the inspecting table 2, it can be noted that the threshold of 0.8 has obtained the best accuracy for PETS09-S2L1, AVG-TownCentre, and TUD-Crossing benchmarks. While for Venice-1, threshold of 0.75 is better. From that, it is recommended to tune the best threshold for each site when installing the system to give the best results.

Computation Time In this part, we are interested to find out the efficiency of using MAS. Table 3 compares the running time results of the proposed approach (i.e., using MAS and computer vision methods) and that without MAS (i.e., using pure computer vision methods) in dealing with AVG-TownCentre benchmark. The accelerated trends of both approaches are plotted in Figure 4. It is clear that employing MAS speeds up the detection and tracking processes. Dividing the problem into multiple tasks that can be handled by various agents provides an ability for parallel computation, and thus to overcome some computer vision limitations.

Table 2: Accuracy results for different values of similarity threshold

Benchmarks	Threshold of similarity				
	0.65	0.7	0.75	0.8	0.9
PETS09-S2L1	58	62	66	**70**	64
AVG-TownCentre	65	70	75	**80**	72
TUD-Crossing	63	68	73	**78**	70
Venice-1	38	40	**45**	42	41

Table 3: Comparing running times on AVG-TownCentre showing MAS effect

# of objects	10	25	50	75	100	150	200	300	400	500	600	700
Without MAS	2	5	10	16	23	35	40	60	80	100	120	140
with MAS	**2**	**4**	**7**	**10**	**15.5**	**23**	**27**	**37.5**	**47**	**55.75**	**63.9**	**71.55**

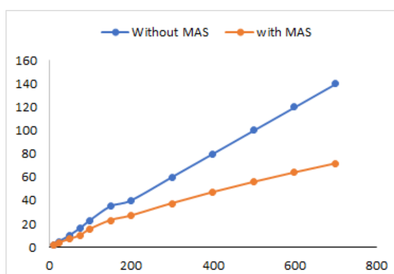


Fig. 4: Running time results

Comparison with other techniques For investigating the proposed approach performance, it is compared to other 14 state-of-the art multi-object trackers in terms of MOTA measure. The comparisons are presented in Table 4. The reported results clarify the superiority of the proposed tracker in dealing with AVG-TownCentre and Venice-1 benchmarks and show the competitive results on the TUD-Crossing benchmark. These results confirm that the employed search agent has assisted the proposed model in predicting and searching for missing persons during the tracking process, and thus achieving good results compared to the results of recent comparative techniques.

Table 4: Comparison between the proposed tracker and other state-of-the art trackers in terms of MOTA% [- indicates that the benchmark has not been tested by the corresponding approach]

Trackers	Benchmarks			
	PETS09-S2L1	AVG-TownCentre	TUD-Crossing	Venice-1
proposed algorithm	**70**	**80**	78	**45**
FFT15	-	36.2	**84.8**	38.6
MPNTrack15	-	54.4	80.1	40
RNN_LSTM	-	13.4	57.2	12.7
SiameseCNN	-	19.3	73.7	22.3
MADRL	-	49.8	79.6	26.5
DeepMP_15		47.3	73.6	36.8
MHTREID15	-	38.4	79.1	32.5
CRFTrack	-	49	80.8	37.3
AP_HWDPL_p	-	28.4	61.3	39.1
STRN	-	27.9	68	39.8
AMIR15	-	36.2	73.7	29.1
HybridDAT	-	29.2	73.3	37.1
INARLA	-	32.1	73	36.4
QuadMOT	-	30.8	72.1	31.8

6 Conclusions and Future Work

From the results above, an efficient multi-person detection and tracking model is presented by combining computer vision and MAS. The model is tested on four real MOT challenge benchmarks. Simulation results demonstrated that the similarity threshold has a significant impact on the performance of the proposed tracker. Besides, adopting MAS has ensured excellent potential of tracking persons in efficient time. In comparison with other recent tracking methods, the proposed method showed high competitive accuracy results.

The future directions of pedestrian tracking will focus on recognizing the entered person, registering her/him, and following him to analyze her/his behavior according to the age, background, level of education, and any other information. This could be used in many applications, including marketing and visitor satisfaction.

References

1. Booranawong, A., Jindapetch, N., Saito, H.: A system for detection and tracking of human movements using rssi signals. IEEE Sensors Journal PP, 1–1 (01 2018)
2. Ray, K.S., Dutta, S., Chakraborty, A.: Detection, recognition and tracking of moving objects from real-time video via sp theory of intelligence and species inspired pso. arXiv preprint arXiv:1704.07312 (2017)
3. Sanchez, S., Rodríguez, S., De La Prieta, F., De Paz, J., Bajo, J.: Multi-agent system for tracking and classification of moving objects. Advances in Intelligent Systems and Computing 373, 63–74 (01 2015)
4. Nguyen, D.T., Li, W., Ogunbona, P.: Human detection from images and videos: A survey. Pattern Recognition 51 (09 2015)

5. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. vol. 1, pp. I–I. IEEE (2001)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005) 2 (06 2005)
7. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Sequential deep learning for human action recognition. In: International workshop on human behavior understanding. pp. 29–39. Springer (2011)
8. Dorri, A., Kanhere, S., Jurdak, R.: Multi-agent systems: A survey. IEEE Access pp. 1–1 (04 2018)
9. Srisamosorn, V., Kuwahara, N., Yamashita, A., Ogata, T., Ota, J.: Human-tracking system using quadrotors and multiple environmental cameras for face-tracking application. International Journal of Advanced Robotic Systems 14, 172988141772735 (09 2017)
10. Choi, J.W., Moon, D., Yoo, J.H.: Robust multi-person tracking for real-time intelligent video surveillance. ETRI Journal 37 (06 2015)
11. Sabha, M., Nasra, I.: Visitor tracking in open cultural places. In: Proceedings of the 4th Hyperheritage International Seminar. HIS.4, Jenin, Palestine (05 2017)
12. Jafari, O.H., Mitzel, D., Leibe, B.: Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras. In: 2014 IEEE international conference on robotics and automation (ICRA). pp. 5636–5643. IEEE (2014)
13. Bouma, H., Baan, J., Landsmeer, S., Kruszynski, C., van Antwerpen, G., Dijk, J.: Real-time tracking and fast retrieval of persons in multiple surveillance cameras of a shopping mall. In: Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2013. vol. 8756, p. 87560A. International Society for Optics and Photonics (2013)
14. Sabha, M., Daoud, J.J.A.: Adaptive camera placement for open heritage sites. In: Proceedings of the International Conference on Future Networks and Distributed Systems. ICFNDS '17, Association for Computing Machinery, New York, NY, USA (2017), https://doi.org/10.1145/3102304.3109813
15. Previtali, F., Iocchi, L.: Ptracking: Distributed multi-agent multi-object tracking through multi-clustered particle filtering. In: 2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). pp. 110–115 (09 2015)
16. Jiang, M., Deng, C., Yu, Y., Shan, J.: Multi-agent deep reinforcement learning for multi-object tracker. IEEE Access PP, 1–1 (02 2019)
17. Rezaee, H., Abdollahi, F.: Average consensus over high-order multiagent systems. IEEE Transactions on Automatic Control 60, 1–1 (11 2015)
18. Marín-Reyes, P.A., Lorenzo-Navarro, J., Castrillón-Santana, M.: Comparative study of histogram distance measures for re-identification (2016)
19. Leal-Taixé, L., Milan, A., Reid, I., Roth, S.: Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942 abs/1504.01942 (04 2015)
20. Bellifemine, F., Bergenti, F., Caire, G., Poggi, A.: Jade—a java agent development framework. In: Multi-agent programming, pp. 125–147. Springer (2005)
21. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The clear mot metrics. EURASIP Journal on Image and Video Processing 2008 (01 2008)