Designing a Visual Sensor Network Using a Multi-Agent Architecture

Federico Castanedo, Jesús García, Miguel A. Patricio and José M. Molina
University Carlos III of Madrid
Computer Science Department
Grupo de Inteligencia Artificial Aplicada (GIIA)
http://www.giaa.inf.uc3m.es
e-mail: {fcastane,jgherrer,mpatrici}@inf.uc3m.es , molina@ia.uc3m.es

Abstract An intelligent Visual Sensor Network (VSN) should consist of autonomous visual sensors, which exchange information with each other and have reasoning capabilities. The information exchanged must be fused and delivered to the end user as one unit. In this paper, we investigate the use of the Multi-Agent paradigm to enhance the fusion process in a VSN. A key issue in a VSN is to determine which information to exchange between nodes, what data to fuse and what information to present to the final user. These issues are investigated and reported in this paper and the benefits of an agent based VSN are also presented. The aim of the paper is to report how the multi-agent architecture contributes to solving VSNs problems. A real prototype of an intelligent VSN using the Multi-Agent paradigm has been implemented with the objective to enhance the data fusion process.

1 Introduction

The main objective of a VSN is to be able to monitor a large geographic area and to receive information of what is happening there [1]. In a VSN with several sensors monitoring an area, data fusion is a main element. The visual sensors could be deployed with overlapped field of views, therefore the information must be fused to provide to the final user a single view [2]. With sensor data correlation and fusion, the VSN can detect targets and track movements in the whole monitoring area. In order to achieve target tracking in a big area, one single visual sensor is not sufficient, the use of several sensors is necessary. Also, some redundancy is needed to carry out the fusion process among all updating sensors.

Multi-sensor data fusion [3] is about combining and relating data from several visual sensors to achieve a better quality than could be achieved by a single visual

This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, SINPROB, CAM MADRINET S-0505/TIC/0255 and DPS2008-07029-C02-02

sensor. Fused data represents an entity in greater detail and with less uncertainty than what is obtainable from any of the individual sources. Combining additional independent and redundant data usually provides an improvement of the results.

A common situation in data fusion is that data sources are spatially or temporally distributed. In complex distributed systems it is impossible, at the design time of the system, to know all the possible interactions between the components of the system. Because communication will occur at unpredictable times and for unpredictable reasons. If we use a classical mathematical modelling approach, in the design phase of the system, all the possible combinations should be programmed. This task could be very complex and requires modelling the error distributions of the visual sensors. Therefore, it is better, from a design point of view, to let the agents at run time decide when to communicate and what to communicate.

VSN are naturally distributed: data sources are spatially distributed, data processing is performed in a distributed manner and the users interested in the result could also be distributed. For these characteristics, VSN definitely falls into the class of potential multi-agent applications.

The multi-agent paradigm has been applied to computer vision in several different works. Zhou et. al [4] use an agent-based approach that uses an utility optimization technique to guarantee that vision tasks are fulfilled. In the embedded multi camera tracking context, the multi-agent paradigm has been also applied. Quaritsch et. al. [5] present an agent-based autonomous multi camera tracking system, providing a soft hand-over in tracking tasks. The basic idea is that agents are instantiated in the embedded device as the tracking object is moving. Several papers using multi-agent systems in distributed sensor networks (without visual capabilities) have been published [6]. However, multi-agent systems applied to visual sensor networks are less explored. VSNs are different from other type of sensor networks because it is necessary to perform image processing tasks in order to obtain valuable information. Berge-Cherfaoui [7] proposed a Multi-Agent approach based on the blackboard model. The blackboard model is one of the first multi-agent communication models and is less flexible than current communication models based on FIPA-ACL [8]. The work of Pavón et. al [9] is very related and provides an useful information on how a multi-sensor system is modelled using agents from a software engineering point of view.

In contrast to other works, this paper focuses on how the multi-agent paradigm is used to develop a VSN enhancing the fusion process. Agents behavior should involve perception, reasoning, communicating, acting and learning in complex environments. In multi-agent VSNs, visual sensing is the mechanism or process whereby the system can be influenced by the environment around it. Therefore, digital image processing techniques must be carried on the agents with the purpose of obtain the information of the object (position, size, velocity and so on). Instead of exchange the raw images between the agents, which implies a lot of communication bandwidth, only relevant information is sent. The main objectives of the proposed multi-agent VSN architecture are:

- To build an open architecture which is easy to scale. We could easily add new agents (with different or same goals) in the multi-agent VSN system.

- A standard based architecture, which would allow us to inter-operate with third part developments.
- Improving the visual sensor processes through reasoning and coordination of the involved visual sensors.

The rest of the paper is organized as follows. In the next section the requirements and objectives of a VSN are presented. Section 3 introduces the multi-agent systems and how they could solve VSN problems. Then, the approach used to implemented the multi-agent VSN is presented. Finally, section 5 concludes the paper with a summary of the major benefits on using a multi-agent VSN.

2 Visual Sensor Networks: Requirements and Objectives

The work on VSN can be divided into two categories, based on visual configuration: (1) spatially overlapping field of view and (2) non-overlapping field of view. In non-overlapping field of view configuration, since there might be significant gaps between the field of views of the cameras, the precision in sensor data fusion may be suffer. Most sensor fusion algorithms assume an overlapping field of view. In this work, an overlapping field of view is also assumed.

Implementing and developing a VSN is a challenging task which involves solving several problems. Firstly, VSN applications must cope with a dynamic and not defined number of detected objects. Therefore, it is mandatory to use algorithms that performs detection, tracking and removing objects from the captured images. In VSN, movement detection and object tracking is performed using several visual sensors connected in a network. VSN are extremely useful because they provide an extended area coverage and could exploit the redundancy of the information. In order to integrate the different parts of a VSN it is necessary to establish: (1) the objectives of each module and (2) how the different modules interact to reason about the context in the global scene.

Some of the open problems to fuse data from sensor in a VSN are: (1) visual sensor configuration, (2) camera calibration, (3) object matching, (4) hand-over, (5) state estimation and (6) occlusion management.

(1) The physical installation of the visual sensors provides a big impact in the performance and economic cost of the system. On one hand, many redundant sensors increase the economic cost. On the other hand, if some areas are not being covered, it could be a risk. Therefore, cover all the area with the minimum visual sensors is an important task. Pavlidis et al. [10] presented an algorithm to cover a parking area with the minimum visual sensors. The basic idea is to increase the visual sensors number with the constraint of having 25%-50% of covered area.

(2) The calibration process (or common referencing) involves projecting the coordinates from the local plane of each visual sensor to global coordinates. The calibration process uses a geometrical projection known as the pinhole model or perspective projection model [11]. The algorithms that reconstruct the 3-D structure from an image, or calculate the object's position in the space need equations to

match the 3-D points with their corresponding 2-D projections. Therefore, we have to calculate what is known as the *extrinsic parameters* (those which relate both 3-D reference systems) and the *intrinsic ones* (which relate the reference system of the camera to the image). This task is known in computer vision as camera calibration process [11]. The pinhole model establishes the mathematical relation between the spatial points and their equivalents in the camera image.

Stein [12] and Lee et al. [13] use the tracking positions of an object and the ground plane restriction to determine the projection's matrix transformation. A different approach is used by Intille and Bobick [14], they propose to use the football field mark lines for calibrate the environment. In our work, the constraint in planar motion (ground plane restriction) is used to estimate a homograph correspondence between the different views and avoid the use of more complex stereo-vision processes. In order to obtain the global coordinates of the local tracks, visual sensors compute the inverse transformation of the middle-bottom point of each object (see Figure Fig. 1).

(3) Object matching involves associating sensor measurements with target tracks. It is known as data association. Data association could be applied at different levels in the sensor environment. In a visual sensor network, the identity of each object must be maintained over the environment being monitored. The most common algorithms used for data association are: nearest neighbor, Multiple Hypothesis Tracking (MHT) [15], Probabilistic Multiple Hypothesis Tracking (PMHT) [16], Probabilistic Data Fusion (PDA) [17], Joint Probabilistic Data Fusion (JPDA) [18] and all their variants.

Fig. 1 Local projection on the ground plane (white point)

(4) In VSN, hand-over takes place when the tracked object moves from one camera's field of view to another camera's field of view. The main reason for doing a hand-over is because the object is going to disappear from the camera's field of view. VSN environments must cope with hand-over in order to allow a global coherence of the objects being monitored in the environment. Quaritsch et. al [5] propose a mobile multi-agent approach in which an agent migrates from one smart sensor to another for solving the hand-over problem. Other authors use a color based approach for matching the detected objects. However this technique is not very robust and fails with illumination changes and with people wearing clothes with similar colors.

(5) State combination from multiple visual sensors is an important problem which many potential applications.

The information exchanged by the visual sensors in the network must be *aligned in time* in the fusion process. Therefore, the information should be time-stamped as soon as possible. The internal clocks of each visual agent are different and perhaps not synchronized. We use the Network Time Protocol algorithm [19] in order to maintain each clock synchronized in a common time basis.

When the information is going to be fused on the fusion node, it must be aligned in time. For this task, an algorithm that interpolates the information to the fused instant is used.

The data fusion techniques involve the combination of estimates obtained with different visual sensors. As we said before, instead of sending the raw images, which imply a lot of communication bandwidth, only tracking information is sending. In the proposed system, each visual sensor performs tracking of the detected persons on their field of view. The tracks of the different overlapped sensor should be fused to obtain a global track. The internal tracking process provides for each detected object X_{T_j}, an associated track vector of features $\hat{X}_{T_j}^{S_i}[n]$, containing the numeric description of their features and state their location, speed, dimensions, etc. as well as the associated error covariance matrix, $\hat{P}_{T_j}^{S_i}[n]$.

Other important problems to solve in a VSN are: (1) data fusion coherence, (3) remove inconsistent information in the fusion process and (3) the coordination of the visual sensors to coverage the area being monitored. Data fusion coherence aims to solve situations in which the detected objects in an overlapped area are different. Thus, it is mandatory a reasoning process between the sensors in order to clarify the situation. Inconsistent information in the fusion process is also related with data fusion coherence and it is carried out by statistical tests between the information [2]. Sensor coordination is necessary in order to provide a mechanism for monitoring an extended area.

3 Multi-Agent Systems and their Application to Visual Sensor Networks

There are different types of multi-agent architectures and can be grouped into: (1) reactive, (2) deliberative and (3) hybrid. We choose the Belief-Desire-Intention agent paradigm (deliberative type), however there are also different ways of modelling multi-agent systems;for example, Corchado and Laza [20] proposed to construct agents using case based reasoning technology. Focusing on the Belief-Desire-Intention (BDI) model of agency [21], an agent is formally specified by its mental state, consisting of its beliefs, desires, intentions and ongoing interaction (with other agents and its environment). Therefore, each agent has its own set of beliefs, desires and intentions. The state of the agent at any given time is a triple (B,D,I), where $B \subseteq Beliefs$, $D \subseteq Desires$ and $I \subseteq Intentions$. An agent's *beliefs* correspond to the information the agent acquires from the environment and the other agents. It represents the knowledge of the state of the world. *Desires* capture the motivation of the agents. A desire represents the state of affairs that the agent would like to bring about. *Intentions* are the basic steps chosen by the agent to achieve its *Desires* and represent the desires which an agent has committed to achieve. *Intentions* constrain the reasoning that an agent requires to perform in order to select the action that will be performed. Practical reasoning involves two important processes: (1) deciding the goals to be achieved and (2) how to achieve them. In our proposed multi-agent VSN architecture, intentions are dropped as the environment evolves or the intention has been already achieved. For example, if the tracking objects disappear from the field of view, the tracking intention will not be achievable. The proposed multi-agent VSN architecture is based on the PRS systems computational model [22] [23], and specifically on the open source multi-agent framework JADEX [24].

The autonomous agents cooperate for the purpose of improving their local information, and cooperation is achieved by message exchange through a declarative high-level agent communication language. FIPA-ACL [8] is the standard language for the communication between the agents. This communication consist of high level speech act theory. Those speech acts can be informing, requesting, accepting, rejecting, believing, and so on. The common objectives of the agents in the multi-agent system are achieved with the help of their interactions. Agents communication takes place using FIPA ACL [8] messages in an asynchronous manner. Therefore, the sender and the receiver do not need to be synchronized in order to communicate. The communications and interactions of the agents provide a natural way by which the information is exchanging in a VSN. Others works, related with agent communication and computer vision, are based on specific communication protocols: (1) the blackboard model [7], (2) the information subscription coordination model [25] and (3) the contract-net protocol [26]; and do not exploit the benefits of a standard communication language. However, FIPA ACL includes an implementation of the contract net protocol.

The use of a BDI multi-agent architecture in a VSN environment provides several advantages with respect to the traditional visual sensor systems:

1. The improvement of accuracy and performance due to cooperation. By means of cooperation agents exchange information. Information redundancy (given by overlapped field of views) could be exploited to enhance the accuracy and performance, i.e., a better tracking accuracy [27] [2].
2. The agents solve problems which are part of a global problem, i.e., most of them perform local tracking using local signal processing, but there is a global view of the tracking which is performed by all of them [28].
3. Sharing information between them, allow agents to correct errors by their ability to make an explicit coordination.
4. In a complex environment with several occlusions, lost of tracks usually occurs, therefore a fusion process is justified to avoid tracking inconsistency.

In figure Fig. 2 a global view of the data fusion architecture is showed. The data fusion algorithm consists on weighting each local sensor track according to the covariance matrix after performing a statistical test and is based on the algorithm of

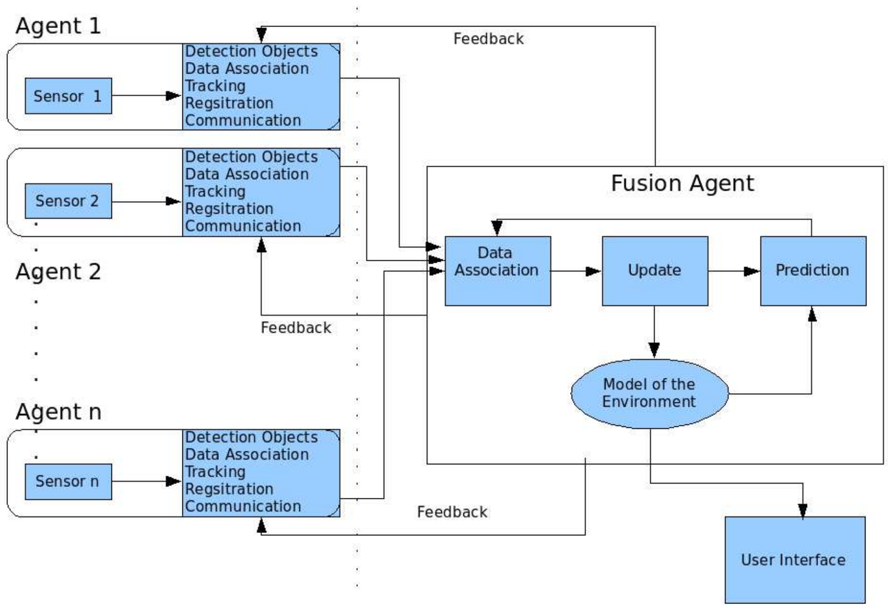

Fig. 2 Multi-Agent VSN architecture with feedback. The figure depicts n sensor agents. Each of them performs: object detection, data association, tracking, registration and communicates with their respective fusion agent. The fusion agent performs data association, updates the (global) model of the environment, make predictions and communicates the information to the interface agent.

the reference [2]. There is a feedback loop between the fused tracks and each local sensor agent. The objective of the feedback information is to allow a reasoning process in each agent in order to correct the local information about the scene. The inverse transformation of the tracked positions is carried in each local agent and a disparity measure is obtained. Then, if the obtained information differs on a defined threshold, the state of the object is changed according to the fused state. Feedback information provides a way to improve the sensor agent's local quality information, using the data fused from all the involved sensor agents. This process is related with the local/global model of the environment. The communication is being performed by means of FIPA ACL messages and provides a communication middleware in the VSN.

In summary, agents in a VSN should reason about the quality of the information which is going to send and receive, negotiate about the targets that are tracking and decide if they take actions to correct or to delete information.

4 Implementation of a Multi-Agent VSN

Using an existing multi-agent platform could provide all common agent tasks as message handling, message encoding/decoding, monitoring and so on. There are several development frameworks of multi-agent platforms, by which some of them are more mature than other. In this work, we choose JADEX [24] as the underlying platform by several reasons. First, it is FIPA compliant, secondly, it is open source and finally it is implemented in Java which allows an easy development, giving the opportunity of obtain experimental results quickly.

The system is divided in three different type of agents, however notice that it could be running N instances of each different type.

1. Sensor agent: this type of agent tracks all the objects and sends the data to a fusion agent. It acquires environmental information through a camera and performs the local signal processing task. The tasks carried out are: object detection, data association, estimation of state and communication with the fusion agent. These tasks are achieved entirely using local data and pixel coordinates and cope with the largest amount of data that can be captured from the video stream. The vision algorithms are based on the OpenCv [29] library. The agent accesses them through Java Native Invocation (JNI) [30]. Each surveillance agent can freely communicate with other agents by exchanging ACL (Agent Communication Language) messages to make the surveillance process more effective.
2. Fusion agent: It fuses the received data from each surveillance agent, which involves: data association, update the model of the environment and predict the evolution of the information. It also sends feedback information to the sensor agents and informs the interface agent.
3. Interface agent: This agent receives the fused data and shows it to the final user, which is also the user interface of the surveillance application. In this graphical user interface the fused tracking information is shown.

The main belief of the sensor agents is the knowledge about the detected objects in their field of view. Sensor agents have the following *desires*: the tracking desires which are performed continuously, the looking desire which allows the agent to observe the environment and the communication desires. Sensor agents have the following *Intentions* (which are generated from the previous desires): tracking, looking the environment, sending new track information (a new object is detected), updating track information (a tracked object change the position) and delete track information (an object disappear).

An indoor evaluation of the proposed architecture was undertaken in our laboratory. Three Sony EVI-100 Pan-Tilt-Zoom cameras were connected to a matrox morphis frame grabber. In this experiment they were used as static cameras. Each PTZ camera is controlled by a surveillance-sensor agent which runs on a dedicated personal computer. There is also a fusion agent which receives each surveillance-sensor agent's tracking information. A record of 300 frames (with a resolution of 768x576 pixels) from one person randomly moving (in a zone of 660cm x 880cm) is performed.

The image sequences of each camera were analyzed by a sensor agent. Inside each sensor agent, the looking interval parameter value was set to 10 milliseconds. The fusion agent data fusion frequency was set to 10 milliseconds, the spatial difference was set to 140 centimeters and the temporal difference was set to 20 milliseconds. For each detected object (track) the centroid was obtained. This position was projected on the ground plane, and then the information in terms of global coordinates was sent to the fusion agent.

5 Conclusions

In this paper the benefits of applying the multi-agent paradigm for modelling a VSN are presented. The specific nature of the multi-sensor visual network, logically and geographically distributed, fit well with the multi-agent paradigm. The use of a classical approach to deal with a visual sensor environment requires to model the sensor's errors which are unknown in most of the visual environments. In contrast, the use of a multi-agent approach allows a distributed reasoning of the information acquired by each visual sensor by means the social ability of each agent. In this work, the standard FIPA ACL communication language is used to share the agent's information. In addition, the use of a multi-agent architecture, based on the FIPA standard, allows the inter-operation with other systems.

References

1. M. A. Patricio, J. Carbó, O. Pérez, J. García, and J. M. Molina. Multi-agent framework in visual sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2007:Article ID 98639, 21 pages, 2007. doi:10.1155/2007/98639.
2. F. Castanedo, M. A. Patricio, J. Garcia, and J.M. Molina. Robust data fusion in a visual sensor multi-agent architecture. *10th International Conference on Information Fusion*, July. 2007.
3. D L. Hall and J. Llinas. *Handbook of MultiSensor Data Fusion*. CRC Press, Boca Raton, 2001.
4. Q. Zhou, D. Parrott, M. Gillen, D. M. Chelberg, and L. Welch. Agent-based computer vision in a dynamic, real-time environment. *Pattern Recognition*, 37(4):691–705, 2004.
5. M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof, and B. Strobl. Autonomous Multicamera Tracking on Embedded Smart Cameras. *EURASIP Journal on Embedded Systems*, 2007(1):35–35, 2007.
6. V. Lesser, C.L. Ortiz, and M. Tambe. *Distributed Sensor Networks: A Multiagent Perspective*. Kluwer Academic Publishers, 2003.
7. V. Berge-Cherfaoui and B. Vachon. A multi-agent approach of the multi-sensor fusion. In *Fifth International Conference on Advanced Robotics.'Robots in Unstructured Environments', ICAR*, pages 1264–1269, 1991.
8. FIPA. Fipa communicative act library specification, 2001.
9. J. Pavón, J. Gómez-Sanz, A. Fernández-Caballero, and J.J Valencia-Jiménez. Development of intelligent multisensor surveillance systems with agents. *Robotics and Autonomous Systems*, 55(12):892–903, 2007.
10. I. Pavlidis, V. Morellas, P. Tsiamyrtzis, and S. Harp. Urban surveillance systems: from the laboratory to the commercial world. *Proceedings of the IEEE*, 89(10):1478–1497, 2001.

11. R. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.
12. G.P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 521–527, 1999.
13. L. Lee, R. Romano, and G. Stein. Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):758–767, 2000.
14. S.S. Intille and A.F. Bobick. Closed-world tracking. *Proceedings of the Fifth International Conference on Computer Vision*, pages 672–678, 1995.
15. D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.
16. R.L. Streit and T.E. Luginbuhl. Maximum likelihood method for probabilistic multihypothesis tracking. *Proceedings of SPIE*, 2235:394, 1994.
17. Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11:451–460, 1975.
18. T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. *19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 19:807–812, Dec. 1980.
19. D. L. Mills. Internet time synchronization: the network time protocol. *IEEE Transactions on Communications*, 39(10):1482–1493, 1991.
20. J. M. Corchado and R. Laza. Constructing deliberative agents with case-based reasoning technology. *International Journal of Intelligent Systems*, 18(12):1227–1241, 2003.
21. M.E. Bratman. *Intentions, Plans and Practical Reasoning*. Harvard University Press, Cambridge, Massachusetts, 1987.
22. M. P. Georgeff and A. L. Lansky. Reactive reasoning and planning. In AAAI Press, editor, *Proceeedings of the Sixth National Conference on Artificial Intelligence (AAAI)*, pages 677–682, 1987.
23. F. F. Ingrand, M. P. Georgeff, and A. S. Rao. An architecture for real-time reasoning and system control. *IEEE Expert,[see also IEEE Intelligent Systems and Their Applications]*, 7(6):34–44, 1992.
24. A. Pokahr, L. Braubach, and W. Lamersdorf. Jadex: Implementing a bdi infraestructure for jade agents. *Search of Innovation (Special Issue on JADE)*, 3(3):76–85, September 2003.
25. Multimedia Object Descriptions Extaction from Surveillance Types. http://www.tele.ucl.ac.be/PROJECTS/MODEST/index.html.
26. T. Graf and A. Knoll. A Multi-Agent System Architecture for Distributed Computer Vision. *International Journal on Artificial Intelligence Tools*, 9(2):305–319, 2000.
27. F. Castanedo, J. García, M. A. Patricio, and J. M. Molina. A multi-agent architecture to support active fusion in a visual sensor network. *Second ACM/IEEE International Conference on Distributed Smart Cameras*, Sept. 2008.
28. F. Castanedo, M. A. Patricio, J. García, and J. M. Molina. Bottom-up/top-down coordination in a multiagent visual sensor network. *IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS*, pages 93–98, 5-7 Sept. 2007.
29. OpenCV. intel.com/technology/computing/opencv/index.htm.
30. JNI. Java native method invocation specification v1.1, 1997.