We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

17

Motion Detection and Object Tracking for an AIBO Robot Soccer Player[1]

Javier Ruiz-del-Solar and Paul A. Vallejos
Universidad de Chile
Chile

1. Introduction

Movement analysis is a fundamental ability for any kind of robot. It is especially important for determining and understanding the dynamics of the robot's surrounding environment. In the case of robot soccer players, movement analysis is employed for determining the trajectory of relevant objects (ball, teammates, etc.).

However, most of the existing movement analysis methods require the use of a fixed camera (no movement of the camera while analyzing the movement of objects). As an example, the popular background subtraction algorithm employs a fixed background for determining the foreground pixels by subtracting the current frame with the background model. The requirement of a fixed camera restricts the real-time analysis that a soccer player can carry out. For instance, a human soccer player very often requires the determination of the ball trajectory when he is moving himself, or when he is moving his head, for making or planning a soccer-play. If a robot soccer player should have a similar functionality, then it requires an algorithm for real-time movement analysis that can perform well when the camera is moving. The aim of this work is to propose such an algorithm for an AIBO robot. This algorithm can be adapted for almost any kind of mobile robot.

The rationale behind our algorithm is to compensate in software the camera movement using the information about the robot body and robot head movements. This information is used to correctly align the current frame and the background. In this way, a stabilized background is obtained, although the camera is always moving. Afterward, different traditional movement analysis algorithms can be applied over the stabilized background. Another feature of our algorithm is the use of a Kalman Filter for the robust tracking of the moving objects. This allows to have reliable detections and to deal with common situations such as double detections or no detection in some frames because of variable lighting conditions.

This chapter is organized as follows. In section 2 we present some related work. In section 3 is described the here proposed motion analysis algorithm for AIBO robots. Experiments

[1] This research was partially funded by Millenium Nucleus Center for Web Research, Grant P04-067-F, Mideplan, Chile.

Source: Robotic Soccer, Book edited by: Pedro Lima, ISBN 978-3-902613-21-9,
pp. 598, December 2007, Itech Education and Publishing, Vienna, Austria

using real video sequences are described and analyzed in section 4. Finally, some conclusions are given in section 5.

2. Related Work

A large literature exists concerning motion analysis in video streams using fixed cameras. As an example, the PETS event is held every year, in which several state-of-the-art tracking and surveillance systems are presented and tested. See for example (Ferryman, 2002) and (Ferryman, 2005). Different approaches have been proposed for moving object segmentation; including frame difference, double frame difference, and background suppression or subtraction. In the absence of any a priori knowledge about target and environment, the most widely adopted approach is background subtraction (Cucchiara et al., 2002). Motion History is another simple and fast motion detection algorithm. According to (Piater & Crowley, 2001), the Motion History and Background Subtraction algorithms have complementary properties, and when possible it is useful their joint use.

Image alignment using gradient descending is one of the most used alignment algorithms. It can be divided into two formulations: the additive approach, which starts with an initial estimation of the parameters, then iteratively finds appropriate parameters' increments, until the estimated parameters converge (Lucas & Kanade, 1981); and the compositional approach, which estimates the parameters using an incremental warp. This last approach iteratively solves the estimation problem using an incremental warp of the images to be aligned with respect to a template. This allows pre-computing the Jacobean more efficiently (Baker & Matthews, 2001). However, the key to obtain an efficient algorithm is switching the role of the image and the template. This leads to the formulation of the inverse compositional algorithm (Baker & Matthews, 2002), where the most computationally expensive operations are pre-calculated, allowing a faster convergence. In (Ishikawa et al., 2002) it was proposed the robust inverse compositional algorithm as an extension to the inverse compositional algorithm, allowing the existence of outliers into the alignment with almost the same efficiency.

Regarding object tracking, Kalman Filtering, Extended Kalman Filtering and Particle Filtering (also known as Condensation and Monte Carlo algorithms) are some of the most common used algorithms. Due to its simplicity, the Kalman filter is still been used in most of the general-purpose applications when the linearity and Gaussianity assumptions are valid.

The here-proposed motion detection and tracking system is based on the described algorithms: background difference and motion history for motion detection, robust inverse compositional algorithm for the image and background alignment, and Kalman filtering for the tracking of moving objects.

3. Proposed Motion Detection System

3.1 System Overview

In figure 1 a block diagram of the proposed system is shown. The system is composed by four main subsystems: *Image Alignment*, *Motion Detection*, *Detection Estimation*, and *Background Update*. In the *Image Alignment* module, the last updated background image (B_{k-1}) and the last frame image (I_{k-1}) are aligned with respect to the current frame image (I_k). The camera motion angles (α_κ) are employed in this alignment operation. Both aligned

images, B_k^* and I_{k-1}^*, respectively, are then compared with I_k in the *Motion Detection* module for determining the current moving pixels. As a result of these comparisons, the *Motion History* and *Background Subtraction* algorithms generate preliminary detections (a set of moving pixels), D_{H_K} and D_{B_k}, respectively. These detections are joined in the Rejection Filter module, and a final set of candidate blobs (moving objects) Det_k is obtained using adjacent moving pixels. The motion detections are analyzed in the *Detection Estimation* module using a Kalman Filter, and the final detections Det_k* are obtained. Finally, the background is updated using B_k^*, I_k and Det_k* (which defines the new foreground pixels) by the *Background Update* module.

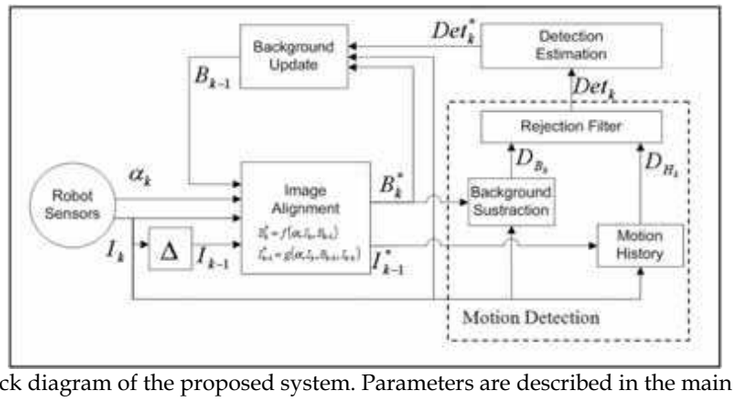

Fig. 1. Block diagram of the proposed system. Parameters are described in the main text

3.2 Image Alignment

The alignment of the last updated background image (B_{k-1}) and the last frame image (I_{k-1}) is implemented using the robust inverse compositional algorithm (Ishikawa et al., 2002). The alignment operation is implemented as a sequence of incremental warps (see section 2). The initial estimation of the warp is calculated based on the camera motion angles (stored in the α_K vector; they correspond to the tilt, pan and roll camera rotation angles). The initial estimated warp is a composition of a rotation followed by a displacement. The angle of rotation is estimated as the variation of the roll angle of the camera, while the displacement D_x/D_y in the X/Y axis corresponds to the pan/tilt angle:

$$\begin{aligned} \alpha_R &= \alpha_{pan2} \cdot \sin(BA - \alpha_{tilt2}) - \alpha_{pan1} \cdot \sin(BA - \alpha_{tilt1}) + \alpha_{roll2} - \alpha_{roll1} \\ Dx &= (\alpha_{pan2} - \alpha_{pan1}) \cdot 180 \\ Dy &= \left((BA - \alpha_{tilt2}) \cdot \cos(\alpha_{pan2}) - (BA - \alpha_{tilt1}) \cdot \cos(\alpha_{pan1})\right) \cdot 180 \end{aligned} \quad (1)$$

where α_R represent the rotation in radians, D_x and D_y represent the displacement in pixels in their respective axis, BA is the angle of the body, and the angles α_{tilt1}, α_{pan1}, α_{roll1}, α_{tilt2},

α_{pan2}, α_{roll2} are the tilt, pan and roll angles of the robot's head in the last and in the current image, respectively, measured in radians.
Then the warp is defined by a set of six parameters as:

$$\begin{aligned} Wx &= (1+P1)\cdot x + P3\cdot y + P5 \\ Wy &= P2\cdot x + (1+P4)\cdot y + P6 \end{aligned} \tag{2}$$

where (W_x, W_y) define the new pixel coordinates which initial coordinates were (x,y). A pure displacement warp has the parameters *P1* to *P4* equals to zero, and the parameters *P5* and P6 equals to the displacement in pixels in the x and y axis respectively. A pure rotation warp has the parameters *P1* to *P4* equals to the rotation matrix, and the parameters *P5* and *P6* equals to zero. Finally, a compound warp of a rotation followed by a translation have the parameters: $P1 = \cos(\alpha_R) - 1$, $P2 = \sin(-\alpha_R)$, $P3 = \sin(\alpha_R)$, $P4 = \cos(\alpha_R) - 1$, $P5 = Dx$, $P6 = Dy$.

For aligning B_{k-1}, the area of the current image (I_k), which has being estimated to overlap the background, is chosen as a template for the algorithm. This preliminary template is divided into nine blocks (sub-images). In each block, the normalized variance of its pixels (intra-block variance), and the normalized variance of the error with respect to the correspondent block in the background (inter-block variance) are calculated. A *variability factor* is computed as the quotient of the intra-block variance and the inter-block variance. The six blocks with the largest variability factors are selected as the final templates for the background alignment. Taking into account the normal camera motion, B_{k-1} and I_k should have different spatial sizes for a correct alignment. In our implementation, B_{k-1} has the same height than I_k, but the double of its width. We will denote the set of parameters defining the warping of the background P_B. The algorithm for obtaining P_B is detailed described in (Ishikawa et al., 2002).

For aligning I_{k-1}, the calculated warp of B_{k-1} is employed as a first approximation. However, given that I_{k-1} and I_k have the same size, the calculated warp has to be actualized with a composition with a prior displacement to achieve the same spatial configuration of the background (I_{k-1} should be translated into background spatial coordinates). Then, the result has to be composed with a post displacement to reach the spatial configuration of the current image (the warped image should be taken back to its original coordinates). Thus, defining *P1* as the set of parameters needed to produce a displacement equal to the last position of I_{k-1} inside the background, and *P2* as the set of parameters needed to produce a displacement equal to the inverse of the estimated final position of I_k inside the background, the warp needed to align I_{k-1} is (the set of parameters defining this warping are P_l):

$$W(\mathbf{x}, \mathbf{P_1}) = W(\mathbf{x}, \mathbf{P_2}) \circ (W(\mathbf{x}, \mathbf{P_B}) \circ W(\mathbf{x}, \mathbf{P_1})) \tag{3}$$

For simplicity on the notation, x denotes both spatial image coordinates. The function $W(x,P^*)$ corresponds to a warping operation over x using the set of parameters P^*.

3.3 Motion Detection

The motion detection module is composed by three algorithms, *Motion History* and *Background Subtraction* for movement detection, and *Rejection Filter* for filtering wrong detections and forming the movement blobs.

3.3.1. Motion History

This module keeps a representation of the motion history in the sequence of frames for each time step *k*. For detecting the motion, the distance in the luminance space between the current image I_k and the last aligned image I^*_{k-1} is calculated, obtaining the difference image DM_k defined as follow:

$$DM_k(\mathbf{x}) = \begin{cases} mIncrement & \text{if } \left| I_k(\mathbf{x}) - I^*_{k-1}(\mathbf{x}) \right| > T_m \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Where *mIncrement* corresponds to a factor of increment in the motion, and T_m corresponds to a motion threshold. DM_k contains the initial set of points that are candidate to belong to the MVOs (Moving Visual Object). In order to consolidate the blobs to be detected, a 3x3 morphological closing (Russ, 1995) is applied to DM_k. Isolated detected moving pixels are discarded applying a 3x3 morphological opening (Russ, 1995). The motion history image MH_k, calculated from DM_k, is then updated as:

$$MH_k = MH_{k-1} * DecayFactor + DM_k \tag{5}$$

Finally, all pixels of MH_k whose luminance is larger than a motion detection threshold (T_h) are considered as pixels in motion. These pixels generate the detection image D_{H_k}. 3x3 morphological closing and opening are applied to D_{H_k}.

3.3.2. Background Subtraction

Foreground pixels are selected at each time *k* by computing the distance between the current image I_k and the current aligned background B^*_k, obtaining D_{B_K} as:

$$D_{B_k}(\mathbf{x}) = \begin{cases} 1 \;\; if \;\; \left| I_k(\mathbf{x}) - B^*_k(\mathbf{x}) \right| > T_p \\ 0 \;\; otherwise \end{cases} \tag{6}$$

In order to consolidate the blobs to be detected, a 3x3 morphological closing is applied followed by a 3x3 morphological opening.

3.3.3. Rejection Filter

Blobs corresponding to neighbor (8-connectivity) moving pixels are built. Both images containing moving pixels, D_{H_K} and D_{B_K}, are used for this computation. For each blob b a movement density MD_b is defined as:

$$MD_b = \frac{\sum_{x \in b} \left| I_k(x) - I_{k-1}(x) \right|}{Area(b)} \quad (7)$$

MD_b measures the average change in the last frame for the blob b. *Ghosts*, defined as groups of pixel that are not moving, but detected as moving because they were part of a MVO in the past, should have a low MD_b, while the MVOs should have a large MD_b. Then, blobs with a small area (area ≤ T_b), with a large area (area ≥ T_s) and blobs with a small movement density ($MD_b \leq T_d$) are considered miss detections and discarded.

3.4 Detection Estimation

Target objects (the MVOs) are tracked by keeping a list with the state of each of them. The state of a given target u includes the position of the center of mass (x_u,y_u), the speed (Vx_u,Vy_u), the area (a_u), and the growing speed (Va_u). For each received movement blob, the area and the center of mass are calculated. These variables are used as sensor observations, and integrated across the different motion detections (the ones coming from D_{H_K} and D_{B_K}), and over time using a first order Kalman Filter (Gong, 2000). This process includes six stages: *prediction*, *observation-target matching*, *update*, *detection of new targets*, *targets elimination,* and *targets merge*. After those six stages, the target state list, whose values are estimated by the Kalman Filter, corresponds to the final motion detections (Det_k*).

Prediction. Using a first order cinematic model, it predicts the state vector for each target based on the last estimated state, and projects the error covariance ahead.

Observation-Target matching. In order to update the targets, it is imperative identifying which observation affects each target. For each observation-target combination, a confidence value is calculated as the probability function given by the Kalman filter for the target evaluated on the observation. For each target, all observations with a confidence value over a threshold T_{t1} are associated with the target. If an observation does not have any associated target, then it is considered as a new target candidate and passed to the *Detection of new targets* stage. For each observation associated with a target, speeds (spatial speed: Vx and Vy, and growing speed Va) are estimated. This estimation is performed using the difference between the target state before the prediction and the observation state, divided by the elapsed time since last prediction.

Update. It computes the Kalman gain, updates the state vector for each target using their associated observations, and updates the error covariance. The targets without associated observations are not updated.

Detection of new targets. All observations without an associated target are considered as new target candidates. Their spatial speed is calculated as the distance from the image border, in the opposite direction of the image center, divided by the elapsed time since the last frame, and their growing speed is set to zero.

Targets elimination. Targets without associated detections in the last two frames are considered as disappeared MVO, and eliminated from the target list.
Targets merge. For each target-target combination, two confidence values are calculated as the probability function given by the Kalman filter for one target evaluated on the other target state. If any of this confidence values is over a threshold T_j, then the two targets are considered equivalents, and the target with the largest covariance (measured as the Euclidian norm of the covariance matrix) is eliminated from the target list.

3.5 Background Update

The background model is computed as the weighted average of a sequence of previous frames and the previously computed background:

$$B_k(\mathbf{x}) = \begin{cases} B_k^*(\mathbf{x}) & if\,(\mathbf{x}) \in DET_k^* \\ \alpha I_k(\mathbf{x}) + (1-\alpha) B_k^*(\mathbf{x}) & otherwise \end{cases} \tag{8}$$

4. Experiments

For the experiments, an AIBO robot using the motion software of the *UChile Kiltros* AIBO soccer team (Ruiz-del-Solar, 2007), configured for allowing just head movements was used. The algorithm runs in the robot in real-time. For analysis purposes, two video sequences were employed. In both, the robot moves its head in an ellipsoidal way, keeping the roll angle of the camera approximately aligned with the horizon. While the robot is moving its head, a ball is moving, once in the same direction of the camera movement, and once in the opposite direction. In figure 2, the different stages of the algorithm while processing the frame 28 of the first video sequence are shown.
The here-proposed motion detection algorithm can be enhanced using additional object information, such as color when detecting moving balls. Thus, in the *Rejection Filter* module, it was implemented a ball color filter applied to the blobs. This color filter uses the average U-V values (YUV color space) from each blob for filtering. If the Euclidean distance between the U-V average value of a blob and the ball U-V value (model) is larger than a threshold *Tc*, then the blob is discarded. This filter decreases significantly the number of false positives errors. It should be stressed that this filter can be applied only after the blobs have been already detected.
The first/second video sequence was 33/37 frames long. 11/14 frames contain a moving ball, but the first appearance of the ball cannot be detected because there is no way to know if the ball is moving or if it is stopped. Thus the relevant information are only 9/12 frames with moving ball, 6/9 of them were successfully detected, which correspond to a successful detections rate of 67%/75%. In table 1, consolidated statistics of the analysis of these video sequences, with and without using the color filter are shown.

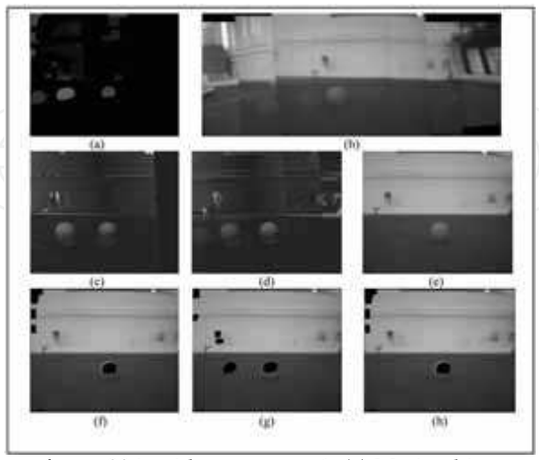

Fig. 2. Process stages at frame 28 in video sequence 1. (a) Motion history representation M_{Hk}. (b) Background model B_k^*. (c) Motion history error image (error intensity values are enlarged for better visualization). (d) Background subtraction error (error intensity values are enlarged for better visualization). (e) Current Image I_k. (f) Motion history detection D_{Hk} in black overlapped to the current image. (g) Background subtraction detection D_{Bk} in black overlapped to the current image. (h) Final detections DET_k*, generated by the Kalman Filter

Sequence number	1		2	
Number of frames	33		37	
Frames with a detectable moving ball	9		12	
Ball color filter	Off	On	Off	On
Frames with successful moving ball detection	6 (67%)	6 (67%)	9 (75%)	6 (50%)
Detections corresponding to moving balls	7	6	9	6
Detections corresponding to ghosts	9	0	7	0
Detections corresponding to other moving objects	10	0	8	0
False detections (excluding ghosts)	296	2	265	5
Total number of detections	322	8	289	11
False detections average by frame, excluding ghosts	8,97	0,06	7.16	0,14

Table 1. Analysis of detections in the video sequence 1 and 2

For a better understanding of the algorithm in figure 3, six frames of the video sequence 1are shown. There is shown the first frame (a), two frames with a ball moving in the opposite direction of the camera movement (b) and (c), and three frames with a ball moving in the same direction of the camera movement (d), (e) and (f). In those frames, there are marks with boxes for the detections, which color represents the detection kind: red for successful detections, blue for fake detections and black for ghost detections. Solid line boxes represent detections after the color filter application; in contrast, dotted line boxes represent detections eliminated by the color filter. In figure 4, six frames of the video sequence 2 are shown. There are shown three frames with a ball moving in the same direction of the camera movement (a), (b) and (c), in those frames the tracking is performed successfully; and three frames with a ball moving in the opposite direction of the camera movement (d), (e) and (f), where the movement is detected partially.

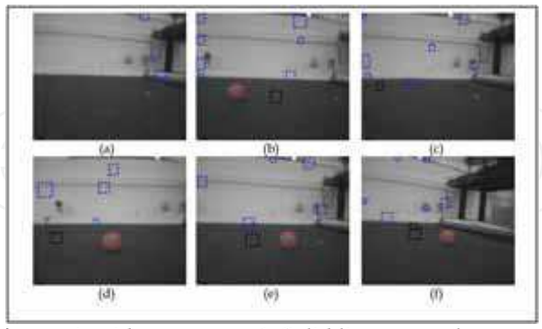

Fig. 3. Six example frames in video sequence 1. Solid line rectangles represent detections after the color filter application; in contrast, dotted line rectangles represent detections eliminated by the color filter. Red boxes denote correct detections; black boxes, ghost detections; and blue boxes, false detections. (a) Frame 1: there is no ball (all movement analysis algorithms need more than 1 frame for making comparisons). (b) Frame 16: the ball and his ghost are detected. (c) Frame 17: the ghost of a disappeared ball is detected. (d) Frame 28: tracking a ball. (e) Frame 29: tracking a ball. (f) Frame 30: tracking a ball

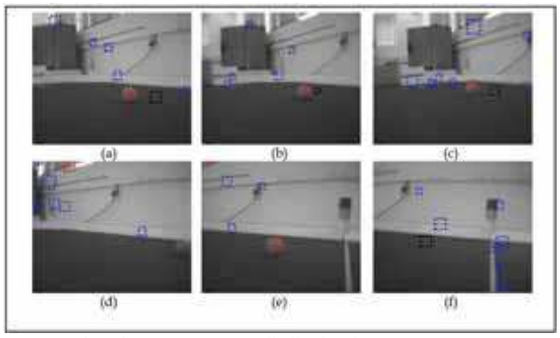

Fig. 4. Six example frames in video sequence 2. Solid line rectangles represent detections after the color filter application; in contrast, dotted line rectangles represent detections eliminated by the color filter. Red boxes denote correct detections; black boxes, ghost detections; and blue boxes, false detections. (a) Frame 13. (b) Frame 14. (c) Frame 15. (d) Frame 24. (e) Frame 25. (f) Frame 26. From frame 13 to frame 15 there is a successful ball tracking. In frames 24, 25 and 26 there is a fast appearance of a ball, in this frames the ball is detected partially

5. Conclusion

In this work we proposed an approach for motion detection and object tracking with a moving camera, with application to robot soccer. In the proposed system, the camera movements are compensated in software by aligning the current frame and the background. Results of the motion detection and tracking of objects in real-world video sequences using

the proposed approach were shown. The system operates in real-time and the relevant moving objects, the ball in this case, are detected and tracked.

In the described system, the images' alignment is achieved using the robust inverse compositional algorithm, which requires an initial estimation of the warp. This initial estimation is obtained from the measured camera motion (robot joints' information). We have developed an improved images' alignment module based on local features (SIFT features), which does not require any information of the camera motion (Vallejos, 2007). This module runs at 5 frames per second. Currently, we are working on reducing the processing time of the module, and in its integration into our motion detection and object tracking system.

7. References

Baker, S. & Matthews, I. (2001). Equivalence and Efficiency of Image Alignment Algorithms, *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, pp I-1090- I-1097, ISBN 0-7695-1272-0.

Baker, S. & Matthews, I. (2002). *Lucas-Kanade 20 years on: A unifying framework: Part 1*, Technical Report CMU-RI-TR-02-16, Carnegie Mellon University, Robotics Institute.

Cucchiara, R.; Grana, C. & Prati, A. (2002). Detecting Moving Objects and their Shadows - An evaluation with the PETS2002 Dataset, *Proceedings of the 3rd IEEE Int. Workshop on PETS*, pp 18- 25, Copenhagen, June 2002.

Ferryman, J. (2002). *Proceedings of the 3rd IEEE Int. Workshop on PETS*, Copenhagen, Denmark, June 2002.

Ferryman, J. (2006). *Proceedings of the 9th IEEE Int. Workshop on PETS*, New York, USA, June 2006.

Gong, S.; McKenna, S. & Psarrou, A. (2000) *Dynamic Vision From Images to Face Recognition*, Imperial College Press, ISBN 1860941818.

Ishikawa, T.; Matthews, I. & Baker, S. (2002). *Efficient Image Alignment with Outlier Rejection*, Technical Report CMU-RI-TR-02-27, Carnegie Mellon University, Robotics Institute, October 2002.

Lucas, B. & Kanade, T. (1981). An Iterative image registration technique with an application to stereo vision, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp 674-679.

Piater, J. & Crowley, J. (2001). Multi-Modal Tracking of Interacting Targets using Gaussian Approximations, *Proceedings of the 2nd IEEE Int. Workshop on PETS*, Hawaii, USA, Dec. 2001.

Ruiz-del-Solar, J.; Guerrero, P.; Arenas, M.; Loncomilla, P.; Fredes, J.; Díaz, G.; Palma-Amestoy, R.; Vallejos, P.; Valenzuela, M.; Dodds, R. & Monasterio, D. (2007). UChile Kiltros 2007 Team Description Paper, *Proceedings of the 2007 RoboCup Symposium* (CD Proceedings).

Russ, J. (1995). *The Image Processing Handbook*, Second Edition, CRC Press inc, Boca Raton, FL, USA, ISBN 0-8493-2516-1.

Vallejos, P. (2007). *Detección y Seguimiento de Personas y Objetos en Movimiento usando Cámaras Móviles*, Master degree thesis, Universidad de Chile, 2007 (in Spanish).

Robotic Soccer
Edited by Pedro Lima

ISBN 978-3-902613-21-9
Hard cover, 598 pages
Publisher I-Tech Education and Publishing
Published online 01, December, 2007
Published in print edition December, 2007

Many papers in the book concern advanced research on (multi-)robot subsystems, naturally motivated by the challenges posed by robot soccer, but certainly applicable to other domains: reasoning, multi-criteria decision-making, behavior and team coordination, cooperative perception, localization, mobility systems (namely omni-directional wheeled motion, as well as quadruped and biped locomotion, all strongly developed within RoboCup), and even a couple of papers on a topic apparently solved before Soccer Robotics - color segmentation - but for which several new algorithms were introduced since the mid-nineties by researchers on the field, to solve dynamic illumination and fast color segmentation problems, among others. This book is certainly a small sample of the research activity on Soccer Robotics going on around the globe as you read it, but it surely covers a good deal of what has been done in the field recently, and as such it works as a valuable source for researchers interested in the involved subjects, whether they are currently "soccer roboticists" or not.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Javier Ruiz-del-Solar and Paul A. Vallejos (2007). Motion Detection and Object Tracking for an AIBO Robot Soccer Player, Robotic Soccer, Pedro Lima (Ed.), ISBN: 978-3-902613-21-9, InTech, Available from: http://www.intechopen.com/books/robotic_soccer/motion_detection_and_object_tracking_for_an_aibo_robot_soccer_player

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen