Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

ELSEVIER

Pattern Recognition Letters 26 (2005) 355–368

Pattern Recognition Letters

www.elsevier.com/locate/patrec

An approach of visual motion analysis

Alberto Sanfeliu [a,*], Juan J. Villanueva [b]

[a] *Institute of Robotics and Department of Automatic Control, Universitat Politècnica de Catalunya,*
Pau Gargallo 5, 08028 Barcelona, Spain

[b] *Computer Vision Centre and Department of d'Informàtica, Universitat Autònoma de Barcelona, Campus UAB, 08193,*
Bellaterra, Barcelona, Spain

Received 22 July 2004; received in revised form 25 October 2004
Available online 22 December 2004

Abstract

In this article, we describe some aspects of the Visual Motion Analysis, where the focus is on the techniques applied in tracking tasks. First we present a Motion Analysis and Recognition (MAR) framework and then we describe methods at two levels of this framework: Image level and 2D level. We explain techniques of motion analysis using single and multiple cues, describing in the last case several cue integration techniques for robust tracking. In order to illustrate the methods, we show several examples of tracking.

In memoriam

Prof. Azriel Rosenfeld was one of the most outstanding pioneers in Computer Vision and it is a great honor for us to dedicate a piece of our work in this special issue. He worked almost in all subjects of the field. In the last years he devoted some time to one of most interesting subject of the area: Visual Motion Analysis. His latest work on this topic has suggested us to write the present paper.

1. Introduction

Autonomous movement is the main function that differentiates the animal kingdom from the vegetal one (Llinás, 2002). To do that, the animals need perception capabilities. When we talk about perception in artificial life, we must talk about sensors that perceive the information of the environment.

In motion analysis, sensors serve for knowing the own movement and the movement of the others. A privileged sensor system for the majority of animal species is the vision one, in particular for humans. More than 1/3 of the human brain cortex is devoted to visual tasks.

[*] Corresponding author. Tel.: +34 93 401 5782; fax: +34 93 401 5750.
E-mail address: sanfeliu@iri.upc.es (A. Sanfeliu).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.10.024

The visual motion analysis is an important area of knowledge that has a lot of applications in many fields, for example in robotics, driver assistance, augmented reality (mobile cameras), traffic control (multiple rigid targets), surveillance, human-machine communication, smart rooms, athletic performance analysis, video conferencing, image storage and retrieval (no rigid tracking objects 2D or 3D, like humans).

We can analyse the vision issues from the point of view of the sensor or the scene. Concerning the vision sensor it can be simple or multiple, fixed or mobile. The scene is formed by the foreground and background, and the scene conditions can change depending if it is an indoor or outdoor scene. For example, often, the illumination can be controlled in indoor scenes, but not in outdoor ones.

The objects (actors if they are humans) which form the scene foreground can be single or multiple and also rigid or no rigid. In this last case, we can analyse the movement in two or three dimensions, using in each case different techniques.

The computer vision techniques that we apply to motion analysis are complex and time consuming. This is especially true in outdoor scenarios with multiple and no rigid targets like human ones. Moreover, real time is mandatory in most of the motion analysis applications. Only the last computer architectures can give a positive answer to these requirements.

In this article we will start explaining a Motion Analysis and Recognition (MAR) framework where we present the main functional modules and their interactions. Then, we describe the tracking using two of the framework modules: Image level and 2D level.

We explain techniques of motion analysis using single and multiple cues, describing in the last case several cue integration techniques for robust tracking. In order to illustrate the methods, we show several examples of tracking.

2. Motion analysis and recognition framework

The study of the computer vision approaches that involve people are known as "look at people" or "dealing with humans". In order to understand and develop techniques to deal with this field we must to identify the functions that take place in motion analysis and recognition. As in the Human Vision System, we can consider that MAR has three kinds of levels: detection, tracking and recognition.

The MAR study implies a lot of difficulties: appearance changes are normal in the scene; foreground and background have to deal with variations in the sensor acquisition conditions, for example due to lighting, occlusions, noise or surface orientation changes. In order to analyse all these factors and be able to analyse the motion, it is necessary to create models at different levels and communicate them in the top–down and bottom–up directions.

Different taxonomies has been presented for Motion Analysis. In the last years, several surveys show different approaches (Gavrila, 1999; Aggarwal and Cai, 1999; Moeslund and Granum, 2001; Wang et al., 2003).

We describe in this article a proposal that tries to include all the main levels that are related with Motion Analysis and Recognition. Fig. 1 shows the proposal.

Signal level forms the "front office" of the system with the world. It is formed by three modules: sensor, actuator and display. Sensor module provides to the system with the information from the scene. We include in this module the hardware equipment, for example the cameras and frame grabbers. Each one of these sensors are characterized by their basic parameters, for example the camera parameters are the focal distance, optical axis, image resolution, frames per second or number de colours. The actuator module can modify these parameters depending on the scene conditions or the upper levels orders. Moreover, the actuator module can act over the world to manipulate the light, switch on an alarm, etc.

Image level: aims to detect and classify objects or actors which form the scene foreground. The rest of scene is the background. A robust detection at this level is essential for the analysis at the 2D level (Moon et al., 2000).

Different features can be obtained from the images using appropriate cues (intensity, colour, texture, shape, ...). Several cues can be used

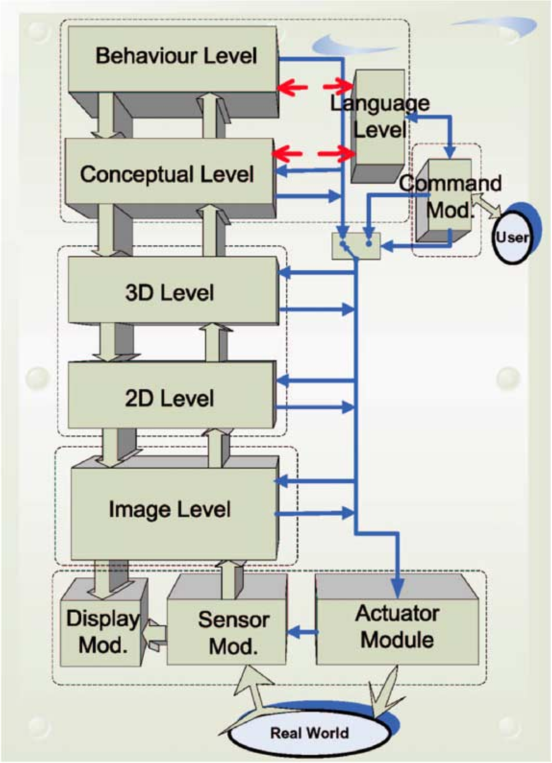

Fig. 1. Motion Analysis and Recognition framework.

together in order to obtain robust results. A particular set of cues is used for each specific task.

To segment the foreground, many techniques can be applied, that means, there is not an universal one. Moreover, there is not exist a general methodology to select the best option for each problem. At present, only the experience can be used to select the best solution in each case.

In motion problems, we can consider two classes of techniques for segmentation: temporal (like optical flow) and spatial ones (like thresholding or statistical methods).

2D level aims to track the objects of interest detected in the Image level. 2D level try to analyse a 2D image from a scene. At this level we work with 2D representations of the objects, or human bodies, and with 2D motion models. The task complexity increases when we try to track groups of people (MacKenna et al., 2000). It is possible to use information coming from the upper 3D level, for example 3D models.

3D level studies the three-dimensional motion of objects and also of human body, particularly parts of it, like limbs or head. There are used models based on sticks, cylinders, or ellipsoids components. The concrete representation is recovered from 2D images. In order to improve the results, some techniques use static and dynamic constraints. Tracking at 3D level is complex and frequently used in an indoor controlled scene (Moon et al., 2001a,b).

Conceptual level analyses the information provided by 2D and 3D levels. This information can be: spatial (like position or orientation); relationship with the environment; pattern of motion. The objective of this level is to associate conceptual interpretation like small, left, fast, walk, to the data of 2D and 3D level.

Behavioural level pretends to recognize what happen in a scene describing a determined conduct in a temporal evolution using description terms. Usually humans and animals have behaviours, but also we can consider that rigid objects like cars or planes can also have behaviours. It is possible to introduce extra knowledge using models or constraints. The inferred information can be structured in a knowledge data base. It is possible to obtain a description of what it is happening at the scene queering the knowledge of the data base.

Language level is the system “front office” with the user. This module serves to communicate, for example, in a natural language, the situation or actions of the actors in the scene. A grammar needs to be created to translate the conceptual terms of static or dynamic actors information into meaningful statements. The quantitative and qualitative information generated in the previous levels are associated with nouns, verbs, adverbs, adjectives and finally with sentences at the Language level.

The system can also generate synthetic image sequences using the textual description. Both synthetic and original sequences can be compared in order to evaluate the system.

3. Single cue approach for tracking

In general, the tracking issue involves two levels in the motion analysis framework: Image level for detecting and/or classifying the objects and the 2D level, to track some of them.

2D level tracks the objects through a sequence of frames. Depending on the applications, the 2D information extracted at this level can be passed to the 3D level or directly to the conceptual one. Besides the connection with the upper levels it is possible to connect it with the low level trough the control pipe selecting the most appropriate cues, changing the segmentation methods or representation structures.

2D level can use information from other higher levels. For example, a 3D object information can be used for a better classification.

In the Image level we can classify the objects based, for example, on shape (blobs, silhouettes), geometry or topological relations (points, edges, curvatures, symmetry, depth), dynamic information (speed, periodic movements) or aspect features (intensity, colour, texture).

Image level extracts different objects from the foreground and performs the classification of them, like people, or human body parts, or other objects like cars.

In 2D level, the tracking performs the matching between each moving object in consecutive frames. The process is based on predicting the next state of the object and evaluate the results according to what it is found in the current image. The state could include information about spatial position, speed, shape, or appearance. At this level motion models are required and several context constraints can be used in order to narrow the search. These constraints could include linear or angular speed or acceleration limits, forbidden areas given by collisions, allowed shapes,...

3.1. Taxonomy

Different categories can be used to study the motion problem, for example, the shape-model versus not shape-model or rigid versus non-rigid objects. The problem is different if the scene is indoor or outdoor, if there is one object or multiple objects, single camera or multiple

The taxonomy for tracking related with our framework model can be presented as follows:

Image level
  Segmentation
  - Background subtraction
  - Optical flow

  Classification
  - Shape based
  - Motion based

2D level
  Tracking
  - Model based
  - Region based
  - Feature based

3D level
  Stick figure
  Volumetric

3.1.1. Image level

3.1.1.1. Segmentation. The aim of segmentation is to extract the parts of the moving foreground. The most known approaches are background subtraction or optical flow.

Background subtraction: tries to detect moving objects in the scene to subtract the current image from another image of reference. This method has problems when the background is moving. For example, the wind can move the leaves of the trees in between consecutive images. Depending on the scene, several methods can improve this situation. Several of them use statistical techniques for representing the pixels of the object of interest. One of them is the mixture of gaussians. Other similar technique to the background subtraction is the subtraction of consecutive frames. Both can be combined in an effective way.

Optical flow: aims to describe coherent motion features (points, edges, blobs) between frames. Optical flow is a robust methodology. It can detect moving objects in an independent way even if the camera is moving too.

3.1.1.2. Classification. The aim of classification is to distinguish between the objects of interest and the rest, all of them moving in the scene. When the moving objects are only the expected type, this problem does not appear. Among the techniques, we can mention the shape-based and motion-based.

3.1.2. 2D level

In 2D level the problems are related with tracking. Tracking an object pretends to match it in consecutive frames. To do this task, the system uses dynamic models based on features like position, velocity, acceleration or appearance based on cues like texture, colour, or shape.

The tracking can be based on models like silhouettes, or regions like blobs, active contour like snakes or based on features like points or lines.

3.1.3. 3D level

3D level information can help to the 2D level. It provides dynamic information about constraints. The most known approaches are based on 3D stick-figure or volumetric models (see Fig. 2).

3.2. Example: Tracking based on appearance as a single cue

A simple case is the study of a problem restricted to a *closed-world* as a region of space and time in which all the present objects are known. If we know the objects and the environment context, we can assume a set of constraints to make easy the tracking task. For example object shape or background colour.

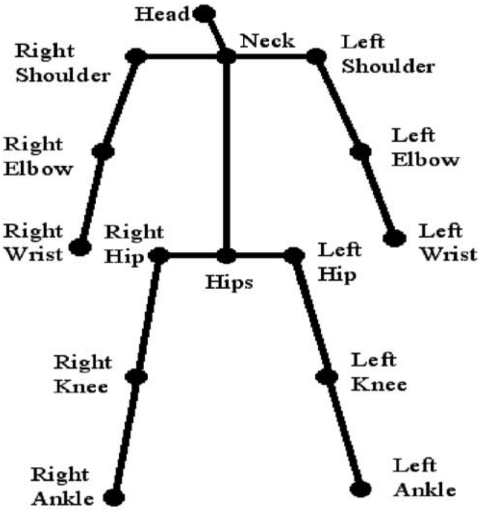

Fig. 2. Stick figure.

We have applied this approach to tracking players in a football match domain (Varona et al., 2000a). The segmentation step has been done assuming that the game field has a uniform green colour. We can extract it with a discriminant analysis algorithm and we have modelled it using a method of maximum likelihood. The Mahalanobis distance has been used to discriminate between players and background (see Fig. 3).

The standard method used in tracking is the Kalman filter. Kalman filter is used to estimated the state over the time. Kalman filter is based on gaussian densities for the state and measurements. This limitation leads to track one unique object. This is an important problem when multiple objects should be tracked in the scene.

In order to solve this problem it is necessary to introduce new approaches, for example, iTrack.

The Bayesian model for temporal state estimation considers the Kalman filter as a particular case. But in general Bayesian approach can be used use for non-linear processes, for example, to track multiple objects.

Let states be denoted by $\boldsymbol{X}_t$ and the measurements denoted by $\boldsymbol{Z}_t$ with

$$Z_t = \{\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_t\}$$

The Bayesian model is,

$$p(\boldsymbol{X}_t|Z_t) = p(\boldsymbol{Z}_t|\boldsymbol{X}_t)p(\boldsymbol{X}_t|Z_{t-1})$$

where

$$p(\boldsymbol{X}_t|Z_{t-1}) = \int_{s_{t-1}} p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})p(\boldsymbol{X}_{t-1}|Z_{t-1})\,\mathrm{d}\boldsymbol{X}_{t-1}$$

this expression defines the likelihood function $p(\boldsymbol{Z}_t|\boldsymbol{X}_t)$ and the dynamics density $p(\boldsymbol{X}_t|X_{t-1})$.

If these densities are gaussians the approach is the Kalman filter. But if the densities are non

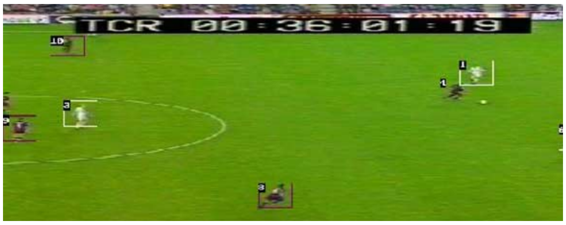

Fig. 3. Tracking football players.

gaussians, like background subtraction or correlations, Kalman filter does not work properly on them.

Based on the image information we complete the estimation algorithm defining functions based on information coming from the Image level and without using any previous model.

Let $\boldsymbol{X}_t = (\boldsymbol{x}_t, \boldsymbol{v}_t, \boldsymbol{f}_t, l_t)$ be the state in time t. Where $\boldsymbol{x}_t$ is the object localization, $\boldsymbol{v}_t$ is the velocity, $\boldsymbol{f}_t$ is the shape of the object, and l_t is the label to identify the object.

The prior density initialise the tracking at the first frame and also is used to initialise the new objects appearing in the scene. In order to locate the objects we define the prior $p(\boldsymbol{x}_t)$ density using background subtraction to the present frame. The objective is to classify the pixels in foreground and background. To do it we use a mixture of gaussians technique,

$$p(\boldsymbol{x}_t) = \sum_b^B \alpha_b G(\mu_b, \Sigma_b)$$

where α_b is a weight. The motion model is used for prediction. We use the velocity to predict next object position,

$$\boldsymbol{v}_t = G(\boldsymbol{v}_{t-1}, \Sigma_v)$$

where Σ_v is covariance of the velocity. We can calculate the new position of the object,

$$\boldsymbol{x}_t = \boldsymbol{v}_t + G(\boldsymbol{x}_{t-1}, \Sigma_x)$$

and the new size and level will be,

$$\boldsymbol{f}_t = G(\boldsymbol{f}_{t-1}, \Sigma_w)$$

$$l_t = l_{t-1}$$

In order to do the prediction we must define a likelihood function. We have selected an object appearance cue based on image data. When an object appears in the scene, the algorithm learns the appearance. This appearance is associated to the object and it is used to make the predictions.

$$L(\boldsymbol{Z}_t|\boldsymbol{X}_t) = \Sigma\Sigma(\boldsymbol{I}_t|\boldsymbol{O}_l)^2$$

$\boldsymbol{I}_t$ is the image patch with the position and size given by the state prediction. $\boldsymbol{O}_l$ is the template corresponding to the object.

Bayesian model performs data association for prediction and estimation. This fact allows to track multiple objects. Using CONDENSATION (Isard and Blake, 1998), we can implement the probabilistic based on a sampling scheme. The density of conditional state $p(\boldsymbol{X}_t|Z_t)$ is represented by a sample set

$$Z_t = \{\boldsymbol{s}_t^n, \pi_t^n, n = 1, \ldots, N\}$$

We can also track new objects which appear in the scene, several samples are generated from the prior information computed in each frame.

The *iTrack* algorithm (Varona et al., 2000b)

1. Choose $\boldsymbol{X}_t$ from $p(\boldsymbol{x}_t)$ [prior initialisation].
2. Choose a base sample $\boldsymbol{X}_t$, by sampling from $p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})$ [temporal prior].
3. Measure the prior using likelihood function to obtain weights π_t.

$$\pi_t^n = L(\boldsymbol{Z}_t|\boldsymbol{X}_t^n)$$

Then normalize the weights

$$\sum_1^N \pi_t^n = 1$$

We can compute the expected object positions by visualizing the weighted samples for each object

$$E(\boldsymbol{X}_t^l|\boldsymbol{Z}_t) = \sum_{s,l} \boldsymbol{X}_t \pi_t$$

Fig. 4 shows the results of a sequence of images.

When a person enters in scene the temporal template is computed and the samples from the prior density are generated. The system tracks the single person based on the temporal template to measure each sample. When a new person appears in the scene a new label is associated to the new object and a temporal template is generated. Finally the system can track the two persons even though occlusions.

The iTrack algorithm is based on image data and it is useful in real applications adapting a proper likelihood function. We have compared our method with two tracking approaches which requires a previous feature extraction step: the Kalman filter and Bayesian filter. Comparison is performed by manually annotating the object position in a sequence. Then we compute the Mean Absolute Error (MAE), the Sum of the Absolute

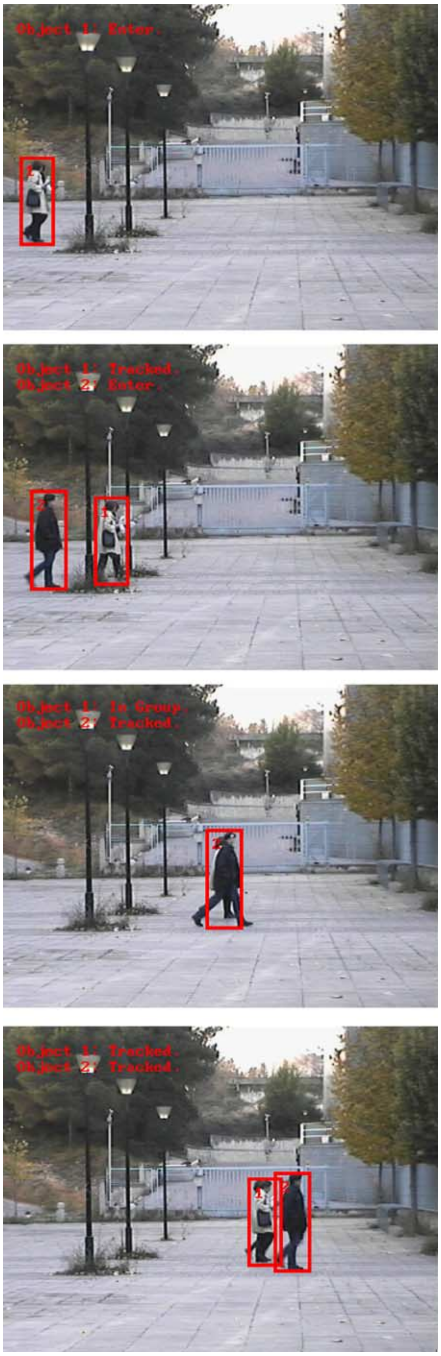

Fig. 4. Results of iTrack algorithm.

Error (SAE) for each method and the uncertainty average. The results are shown in Table 1.

Table 1
Comparative results among different methods

	SAE	MAE	Uncert. Avg
Kalman	914.00	7.25	8.18
Bayesian	680.18	5.39	10.55
iTrack	247.29	1.96	7.55

4. Integration of cues for tracking

Cue integration (fusion) schemes has become an important issue in the last years (Fayman et al., 1999; Wu and Huang, 2004; Kragic and Christensen, 2001; Moreno and Sanfeliu, 2004) to provide useful and high quality information in dynamic environments, as biological perception systems do. Biological systems use dynamic perception (mainly visual) to improve robustness, to overcome modifications of the environment changes and adapt themselves. Studies in the area of active vision has also shown that the use of integration of vision cues can also eliminate some ill-posed problems in several computer vision problems (Aloimonos et al., 1988) as well as, to overcome some problems due to occlusion between objects.

In scene images, the changes of the environment conditions are usually related to the changes of the illumination (for example, shadows, surface reflectance, object geometry and object position), the number and type illumination sources, the background characteristics, the number of moving objects and the occlusion of objects. The robustness is usually related to the figure foreground segmentation (detection of the target), matching across images, inadequate modelling of motion and failure of one of several sensors.

In order to integrated different cues we should select the best ones for each application, these must be selected using some criteria. For example, for run time tracking, the criteria is to use very simple cues that can be computed at frame rate. The typical cues are colour, edges, features from motion, intensity variation, texture and stereo-vision.

4.1. Taxonomy

The information of the cues can be integrated in the different ways, depending of the objective pursued. Basically there have been proposed three types of integration (fusion) of cues:

- One level fusion:
  - Voting scheme
  - Bayesian fusion
  - Fuzzy logic fusion
  - Democratic scheme
- Hierarchical fusion
- Co-operation of cues for fusion

One level fusion is related to the mechanism of integration where all the cue data is collected in parallel and the decision is made by combining them. For example, colour, disparity and texture are collected, each individual cue is normalized in the interval $[0, 1]$ and then the decision is taken by a weighted summation. In the hierarchical fusion the decision is taken at different levels up to arrive to the final decision. In co-operation fusion, several cues co-operate to validate the partial results.

In the following subsection the different techniques will be described.

4.1.1. One level fusion

We describe four well known techniques that have been applied successfully for cue integration. Let us start for the voting scheme which has been applied to reliable systems in networks, microprocessors and aerospace.

4.1.1.1. Voting scheme. Voting methods (Parhami, 1994), deal with n input data objects k (the cues), with n associated non-negative votes (the weights of the cues) w_i, and the objective is to compute the output y and its vote v such that y is "supported by" several input data objects with votes totalling v, where $\mathbf{v}$ satisfies a condition which is associated with the desired threshold or plurality voting scheme.

In cue fusion, the voting scheme enables to increase the reliability of the information of the cues, where the reliability of each individual cue varies significantly over time. One important advantage of the voting scheme is that this mechanism is "model free" with respect to the individual cues, that is, each single cue has the same range of influence in the voting process since they, individually, are bounded in the interval $[0; 1]$.

If we consider an estimation/classification space, θ (the class domain), then v is the mapping v_i: $\theta \rightarrow [0; 1]$. If each of the n cue estimators (v) produces a binary vote for a single class, then a set of thresholding schemes can be used:

$$\begin{aligned} &\text{Unanimity:} && \sum \upsilon_i(\theta) = n \\ &\text{Byzantine:} && \sum \upsilon_i(\theta) > 2n/3 \\ &\text{Majority:} && \sum \upsilon_i(\theta) > n/2 \end{aligned}$$

where θ represents a specific class. If each cue estimator is allowed to vote for multiple classes, then we can use the consensus voting, where the maximum vote is used to designate the winning class θ

$$\theta' = \arg\max \lambda(\theta)$$

where $\lambda(\theta)$ is a combination method, for example it could be a linear combination

$$\lambda(\theta) = \sum_{i=1}^{n} w_i \, * \, \upsilon_i(\theta)$$

A more general class of voting schemes can be defined using the *m-out-of-n* voting scheme. In this case, if we take a confidence value for each one of the cues, we can consider that if a cue does not have enough confidence, then it must be not included in the fusion process. Usually, a cue estimator can give a vote for a given class θ, if the output of the estimator is greater than zero.

A probabilistic integration scheme using Bayesian method and voting scheme can be presented in the following way. Let the likelihood of the observations $Z_{i,k}$ from cue k at pixel i given the model $M_{j,k}$ of layer j (there are several layers, for example foreground, background, etc.) be denote by $p_{i,k}(Z_{i,k}|M_{j,k})$. Then the posterior probability of layer j can be formulated using the Bayes' Rule

$$p_{i,k}(j|Z_{i,k}) = \frac{p_{i,k}(Z_{i,k}|M_{j,k})p(j)}{\sum_j p_{i,k}(Z_{i,k}|M_{j,k})p(j)}$$

where $p(j)$ is the marginal probability of layer j that can be used to express belief concerning the size of the foreground relative the background. Then the combination of the cues can be presented as before by

$$\lambda_i(j) = \sum_{k=1}^{n} w_k * p_{i,k}(j|Z_{i,k})$$

4.1.1.2. Bayesian fusion. The voting scheme presented before using Bayes' Rule is invalid from a probabilistic point of view. In order to obtain the probabilistic integration scheme we make the assumption that the observations from cues are independent, in this way the total likelihood of observations given the combined model $M_j = (M_{j,1},\dots,M_{j,k})$ over all cues k for layer j at pixel i is,

$$p_i(Z_i|M_j) = \prod_k p_{i,k}(Z_{i,k}|M_{j,k})$$

and a posterior estimate of layer membership is

$$p_i(j|Z_i) = \frac{\prod_k p_{i,k}(Z_{i,k}|M_{j,k})p(j)}{\sum_j \prod_k p_{i,k}(Z_{i,k}|M_{j,k})p(j)}$$

This model of Bayesian fusion presents some important differences with respect to the voting scheme described before using Bayes' Rule. The observations from the different cues are combined before the layer membership meanwhile, in the previous one, first the layer membership for each cue were computed and second, the results were combined. By looking the last equation, it can be seen that the completely uncertain cues have not effect on the posterior estimate, in contrast with voting where the scores for each layer were blurred (see Hayman and Eklundh, 2002 for more detail explanation).

4.1.1.3. Fuzzy logic fusion. If instead of using the Bayesian' Rule we use the Fuzzy logic' Rule, then the problem have to be rewritten as follows. Let F fuzzy set, defined as

$$F = \{(\boldsymbol{x}, \boldsymbol{\mu}_F(\boldsymbol{x})|\boldsymbol{x} \in \theta\}$$

where θ denotes the universe of discourse for the set F and $\boldsymbol{x}$ (for example position) is an element of θ. The membership function μ_F gives a membership value $\mu_F(\boldsymbol{x})$ for each element $\boldsymbol{x}$: μ: $\theta \rightarrow [0;1]$. The composition operator can be the min–max operator defined by Zadeh (1973).

$$\boldsymbol{\mu}_{R_1 \circ R_2 \circ \cdots \circ R_n} = \max\{\min(\boldsymbol{\mu}_{R_1}, \boldsymbol{\mu}_{R_2}, \dots, \boldsymbol{\mu}_{R_n})\}$$

4.1.1.4. Democratic scheme. The democratic scheme (Triesch and von der Malsburg, 2001) use a similar probabilistic integration scheme of the Bayesian method and voting scheme, but allowing adapting the internal parameters and the weights of the cues. In this case the fusion function is as follows,

$$p_{c,t}(x) = \sum_{k=1}^{n} w_{k,t} * p_{j,t}(x)$$

where $p_{c,t}(\boldsymbol{x})$ is denominated the saliency map (the probability distribution of the global result for $\boldsymbol{x}$ at time t) into which the different cue probability distributions $p_{k,t}(\boldsymbol{x})$ are fused to produce the final result. For example, for tracking purposes, the final result is an estimated state of $\boldsymbol{x}$, the position of the tracked object.

In this scheme, the parameters can be updated by feeding back the global result to the individual cues, and the weights can also be updated by using a quality measure, $q_{k,t}$. This quality measure is obtained by comparing the two probability distributions of $p_{c,t}(\boldsymbol{x})$ and $p_{k,t}(\boldsymbol{x})$, and then the more similar the distribution $p_{k,t}(\boldsymbol{x})$ is to the global result, the higher is the rating of the underlying cue. Finally the weights are adapted as follows

$$w_{k,t} = (1 - \boldsymbol{\tau}) \cdot w_{k,t-1} + \boldsymbol{\tau} \cdot q_{k,t-1}$$

where τ is an adaptation parameter.

4.1.2. Hierarchical fusion

There are other types of fusion schemes, for example hierarchical fusion, which instead of doing the fusion in one single level, the fusion is done through several hierarchical levels. For example, in the work of (Kähler et al., 2004), a hierarchical sensor data fusion is presented which use probabilistic cue integration for robust 3D object tracking. The objective is to solve the fusion step, by hierarchical fusing the information of the different sensors and different information sources (cues) derived from each sensor. In the first level, the cues of each camera are fused using a democratic scheme and then, in the second level all the fused information are combined to estimate the 3D position of the object.

4.1.3. Co-operation of cues for fusion

An interesting fusion approach is based on the co-operation among several cues to fuse the information for a specific goal. In this case, the objective is to combine cues that in co-operation can improve the performance of each independent cue. There have been proposed different methods of fusion, but we will explain briefly only one of them.

4.1.3.1. Fusion colour and stereovision. We will explain a specific case where the objective was to track objects in real time (30 ms/image) (Moreno et al., 2002) by the co-operation of colour histogram and stereovision (Image level), and using Kalman filter (2D level). The Kalman filter is used for estimating the 3D position of the object to track (in this case a human face) and colour and stereovision are the cues. The method work as follows: The process captures a pair of synchronized stereo colour images and then, the left image is fed into the colour module. By using the information of the previous state about the position on the image and the scale of the head (modelled as an ellipse), the system computes the position of the head in the new image. The search is done by maximizing an intersection function between the colour histogram of the new head candidate and a model histogram. The later is updated by taking into account the colour histogram of the best candidate. The co-operation between colour histogram and stereovision is done at the level of the region to consider. The stereovision computes the distance of the face and this distance is used to create the size of the ellipse to consider. In this ellipse region the system analyse the colour. On the other way, the stereovision is only done in the ellipse considered in the previous image frame and for this reason the system can run in real time. Fig. 5 shows the head tracking system, Fig. 6 presents the comparison with and without fusion and finally in Fig. 7 it is shown the computing time for a Pentium III.

4.2. Example: Fusion of colour and shape for object tracking under varying illumination

We will describe an integration technique that fuses colour and shape for object tracking using co-operation of cues (Moreno-Noguer et al., 2003). The technique is based in the particle filter (Isard and Blake, 1998).

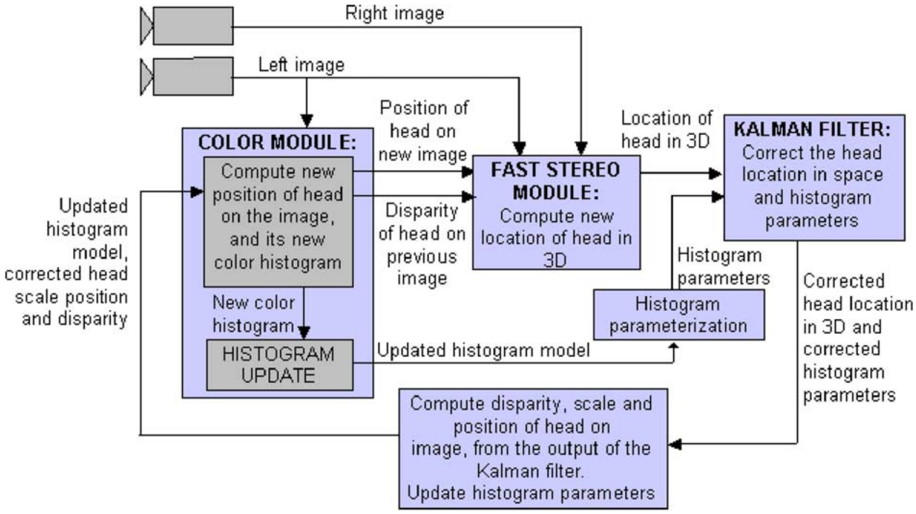
Right image
Left image
COLOR MODULE:
Compute new position of head on the image, and its new color histogram
New color histogram
HISTOGRAM UPDATE
Position of head on new image
Disparity of head on previous image
FAST STEREO MODULE:
Compute new location of head in 3D
Location of head in 3D
KALMAN FILTER:
Correct the head location in space and histogram parameters
Histogram parameters
Histogram parameterization
Updated histogram model
Corrected head location in 3D and corrected histogram parameters
Updated histogram model, corrected head scale position and disparity
Compute disparity, scale and position of head on image, from the output of the Kalman filter.
Update histogram parameters

Fig. 5. General scheme of fusion stereovision and colour.

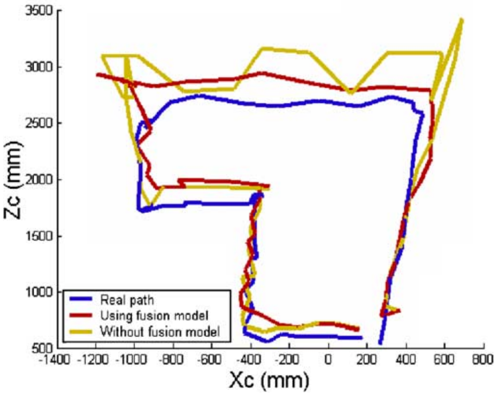

Fig. 6. Comparison between trials made using and not using fusion.

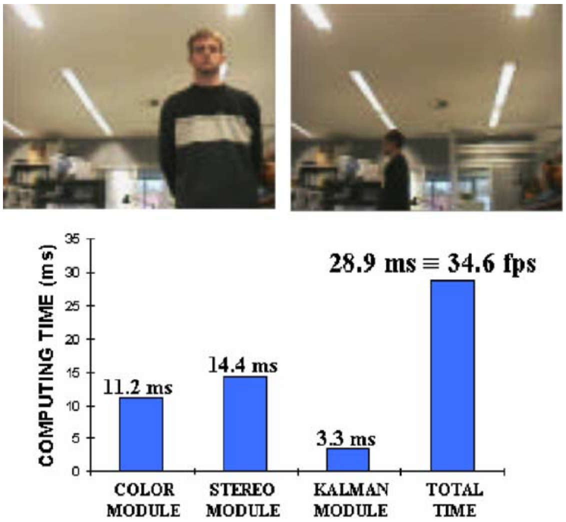

Fig. 7. Analysis of the time computing.

The main features of the method are the ability to adapt shape deformation and track an object under varying illumination conditions. The method uses two cues, the colour change of the object and the shape deformation of the object, although at present, we only allow affine deformations. The basic idea of the method is to use the particle filter as a probabilistic framework to track the colour in the colour space (Image level). The second cue, the contour, co-operates with the previous one to detect the best candidate in the image space (Image level) proposed by the motion analysis (2D level).

In other words, using the predictive filter, multiple estimates of the object colour distribution are formulated at each iteration. These estimates are weighted and updated taking into account the object shape, enabling the rejection of objects with similar colour but different shape that the target. Finally, the best colour distribution is used to segment the image and refine the object's contour.

4.2.1. The tracking algorithm

The algorithm follows the steps of the filter of particles.

The method is based on tracking the object colour distribution C_t, that at time t is the collection of image pixels colour values I_t that belong to the target. In the RGB colour space, the object colour distribution is modelled as

$$X_t = (m_t^{\mathrm{T}}, \lambda_t^{\mathrm{T}}, \theta_t, \phi_t)^{\mathrm{T}}$$

where m_t is the centroid, λ_t are the magnitudes of the principal components and θ_t, ϕ_t the angles centered at m_t, of C_t. Fig. 8 shows the object colour distributions at time $t-1$ and t.

At time t, a set of N samples $s_{t-1}^{(n)}$ $(n = 1,\ldots,N)$ of the form $\boldsymbol{X}$, parameterizing N colour distributions $C_{t-1}^{(n)}$ are available. Each distribution has an associated weight $\pi_{t-1}^{(n)}$, and the whole set represents an approximation of the a posteriori density function $p(X_{t-1}|Z_{t-1})$ (see Fig. 9), where $\boldsymbol{Z}_{t-1} = \{z_0,\ldots,z_{t-1}\}$ is the history of measurements.

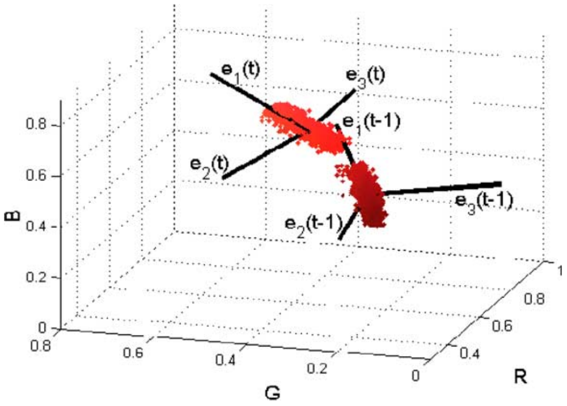

Fig. 8. Object colour distribution at time t and time $t-1$.

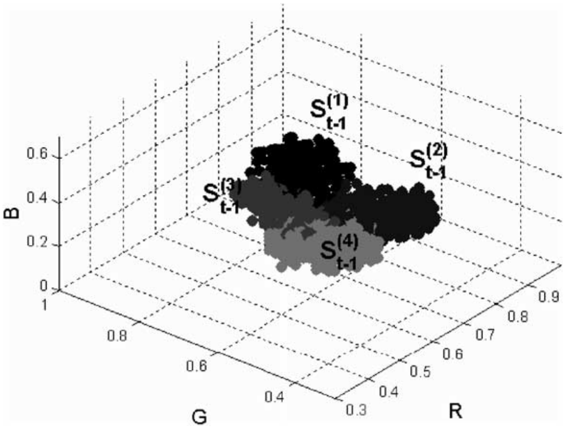

Fig. 9. All the colour distributions of the set $S_{t-1}^{(n)}$.

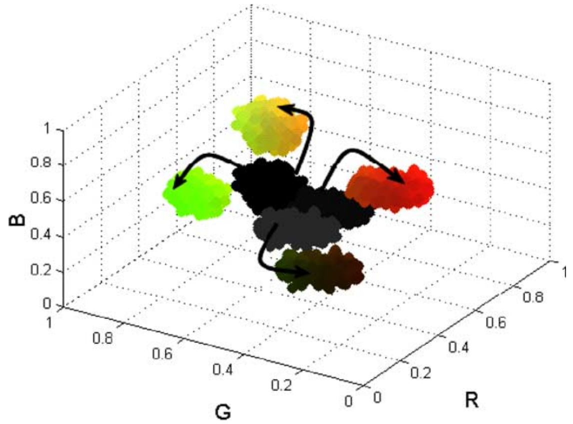

Fig. 10. Sampling and probabilistic propagation from colour distributions of Fig. 9.

Step 1: Sampling from $p(X_{t-1}|Z_{t-1})$. In order to estimate $p(X_t|Z_t)$ the next step is sampling with replacement N times the set $s_{t-1}^{(n)}$, where each element has probability $\boldsymbol{\pi}_{t-1}^{(n)}$ of being chosen. This will give us a new set of colour distribution parameterizations, $s_{t-1}'^{(n)}$.

Step 2: Probabilistic propagation of samples. Each sample $s_{t-1}'^{(n)}$ of the set is propagated according to a dynamic model and the result can be seen in Fig. 10.

Step 3: Fusion of colour distribution and shape. Following the particle filter, in this step, each element $s_t^{(n)}$ has to be weighted according to some measured features. In our case we use the shape information in order to assign higher weights to the samples $s_t^{(n)}$ generating "better" segmentations of the tracked object. These segmentations are done using the histograms of the propagated colour distributions $C_t^{(n)}$ (see Figs. 11, 12). The weight assigned to $s_t^{(n)}$ is computed as follows: $\pi_t^{(n)} = \mathrm{e}^{-\frac{\rho^2}{2\sigma^2}}$ where

$$\rho = \mu_1(1 - \Phi_{\text{affine}}) + \mu_2(1 - \Phi_{\text{area}}) + \mu_3(1 - \Phi_{\text{quality}})$$

where Φ_{affine} (the affine similarity) measures the similarity between the image edges and a snake adjusted to the contour, Φ_{area} evaluates the difference between the area of the snake and the

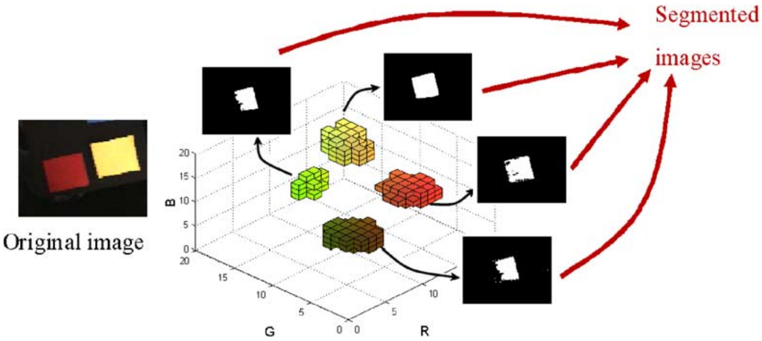

Fig. 11. Segmentation using the colour distribution.

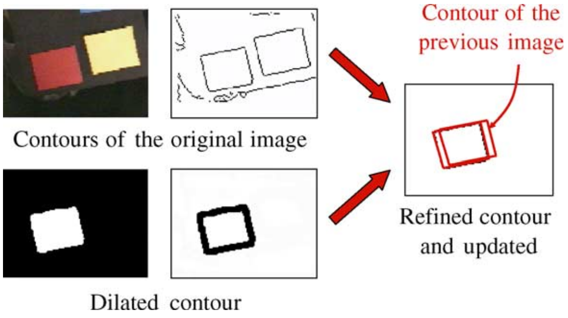
Contour of the previous image
Contours of the original image
Refined contour and updated
Dilated contour

Fig. 12. Computing the cost to adjust the shape to the object image.

predicted area and Φ_{quality} penalize those segmentations of low quality (the ones that have holes into the segmented area). The three measures return a value in the range [0, 1].

Step 4: Contour updating. The last step consists in refining the fitting of the object contour, in order to obtain the points of the along the snake adjusted to the contour.

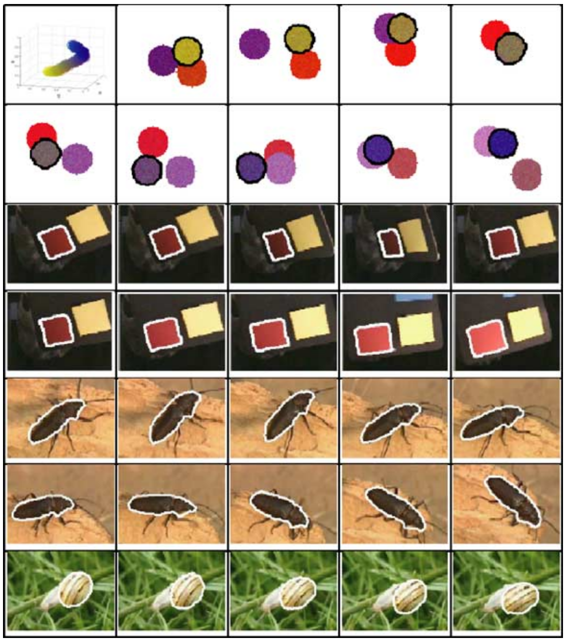

Fig. 13. Four experiments: tracking circles, rectangles, cockroach and snail in different illuminations conditions, background and shape.

The results of these processes can be seen in Fig. 13 for different tracking experiments (tracking circles that move around and change randomly their colour; tracking a coloured rectangle with change of surface orientation and illumination; tracking a cockroach in and outdoor environment; and tracking a snail in a confusing environment).

5. Conclusions

Motion analysis has become an important issue in man-machine communication, robotics, traffic control and many other applications. In the present paper, we present a general framework for motion analysis and we analyse one of the layers, the 2D level, which deals with 2D motion analysis. We explain the taxonomy of techniques for tracking objects, human beings and animals using one and multiple cues and we explain how to obtain robust methods for tracking using integration of cues when the environment conditions change. We also explain several examples of tracking human, objects and animals in diverse situations.

References

Aggarwal, J.K., Cai, Q., 1999. Human Motion Analysis: A review. Computer Vision and Image Understanding 73, 428–440.

Aloimonos, Y., Weiss, I., Bandyopadhyay, A., 1988. Active vision. Int. J. Comput. Vision 1, 333–356.

Fayman, J.A., Pirjanian, P., Christensen, H.I., Rivlin, E., 1999. Exploiting process integration and composition in the context of active vision. IEEE Trans. Syst., Man Cybernetics—Part C: Appl. Rev. 29 (1).

Gavrila, D.M., 1999. The visual analysis of human movement: A survey. Comput. Vision Image Understanding 73, 82–98.

Hayman, E., Eklundh, J.O., 2002. Probablilistic and voting approaches for cue integration for figure-ground segmentation. Eur. Conf. Comput. Vision (ECCV) (3), 469–486.

Isard, M., Blake, A., 1998. Condensation-conditional density propagation for visual tracking. Int. J. Comput. Vision 28 (1), 5–28.

Kähler, O., Denzler, J., Triesch, J., 2004. Hierarchical sensor data fusion by probabilistic cue integration for robust 3D object tracking. In: 6th IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 28–30.

Kragic, D., Christensen, H.I., 2001. Cue integration for visual servoing. IEEE Trans. Robotics Automation 17 (1).

Llinás, R., 2002. I of the Vortex: From neurons to self. The MIT Press.

MacKenna, S., Jabri, S., Duric, Z., Rosenfeld, A., Wechsler, H., 2000. Tracking Groups of People. Comput. Vision Image Understanding 80, 42–56.

Moeslund, T.B., Granum, E., 2001. A survey of computer vision-based human motion capture. Comput. Vision Image Understanding 81, 231–268.

Moon, H., Chellapa, R., Rosenfeld, A., 2000. Optimal shape detection. Int. Conf. Image Process. (ICIP), 885–888.

Moon, H., Chellapa, R., Rosenfeld, A., 2001a. Tracking of Human Activities Using Shape-Encoded Particle Propagation. Int. Conf. Image Process. (ICIP), 357–360.

Moon, H., Chellapa, R., Rosenfeld, A., 2001b. 3D Object Using Shape-Encoded Particle Propagation. Int. Conf. Comput. Vision (ICCV), 307–314.

Moreno-Noguer, F., Andrade-Cetto, J., Sanfeliu, A., 2003. Fusion of colour and shape for object tracking under varying illumination. Pattern Recognition and Image Análisis, First Iberian Conference; IbPRIA 2003. Lecture Notes in Computer Science, LNCS 2652.

Moreno-Noguer, F., Tarrida, A., Andrade-Cetto, J., Sanfeliu, A., 2002. 3D real-time head tracking fusing colour histograms and stereovision. In: Proceedings of the 16th International Conference on Pattern Recognition, Quebec.

Moreno-Noguer, F., Sanfeliu, A., 2004. Integration of shape and multihypothesis Fischer colour model for figure-ground segmentation in non-stationary environments. In: Proceedings of the International Conference on Pattern Recognition, ICPR'04, Cambridge.

Parhami, B., 1994. Voting algorithms. IEEE Trans. Reliability 43 (4).

Triesch, J., von der Malsburg, C., 2001. Democratic integration: self-organized integration of adaptive cues. Neural Computation. 13 (9), 2049–2074.

Varona, X., Gonzalez, J., Roca, X., Villanueva, J., 2000a. iTrack: Image-based Probabilistic Tracking People. Int. Conf. Pattern Recog. 2, 1122–1125.

Varona, X., Pujol, X., Villanueva, J., 2000b. Visual Tracking in applications Domains. Pattern Recognition and Applications, IOS Press.

Wang, L., Hu, W., Tan, T., 2003. Recent Developments in Human Motion Analysis. Pattern Recog 36 (3), 585–601.

Wu, Y., Huang, T.S., 2004. Robust visual tracking by integrating multiple cues based on co-inference learning. Int. J. Comput. Vision 58 (1), 55–71.

Zadeh, L., 1973. Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. Syst., Man, Cybern. SMC-3 (1), 28–44.