SWARMTRACK: A PARTICLE SWARM APPROACH TO VISUAL TRACKING

Luis Antón-Canalís[1], Elena Sánchez-Nielsen[2], Mario Hernández-Tejera[1]
[1]*Instituto de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería. Campus Universitario de Tafira, 35017 Gran Canaria, Spain.*
[2]*Departamento de E.I.O. y Computación, 38271 Universidad de La Laguna, Spain*

Keywords: Computer Vision, Real Time Object Tracking, Swarm Intelligence.

Abstract: A new approach to solve the object tracking problem is proposed using a Swarm Intelligence metaphor. It is based on a prey-predator scheme with a swarm of predator particles defined to track a herd of prey pixels using the intensity of its flavours. The method is described, including the definition of predator particles' behaviour as a set of rules in a Boids fashion. Object tracking behaviour emerges from the interaction of individual particles. The paper includes experimental evaluations with video streams that illustrate the robustness and efficiency for real-time vision based tasks using a general purpose computer.

1 INTRODUCTION

Tracking moving objects is a critical task in computer vision, with many practical applications such as vision based interface tasks (Turk, 2004), visual surveillance (Sánchez-Nielsen, 2005a) or perceptual intelligence applications (Pentland, 2000).

Template based approaches track a target through a video by following one o more exemplars (templates) of the visual appearance of the object.

The template tracking problem has been classically formulated as a search problem of a pattern in the current frame of the video stream that matches the exemplars as closely as possible. Several solutions have been proposed in this sense to deal with the problem. At present, there are still obstacles in achieving all-purpose and robust tracking systems. Different issues must be addressed in order to carry out an effective tracking approach: (1) Dynamic appearance of deformable or articulated targets, (2) Dynamic backgrounds, (3) Following different target motions without restriction, (4) Changing lighting conditions, (5) Camera motion and (5) Real-time performance.

In this paper, a new approach is proposed. The solution is based on a Swarm Intelligence paradigm and, particularly, on focusing the tracking problem under the eyes of a predator-prey metaphor. In our tracking context the template is a sample of prey pixels that supply the *scent* of preys to be tracked to a swarm of predator particles. Then, using a prey scent similarity principle, each predator particle will track its prey. As a result, the tracking of the object will be an emergent property of the Swarm of Particles, where tracking behaviour appears thanks to a set of individual and group behaviour rules.

In the next section, a review of the tracking problem is included. In section 3, a presentation of Swarm Intelligence is detailed. Sections 4 and 5 describe the proposed method. Section 6 includes experimental evaluations of the proposal with video streams in different contexts and finally, section 7 discusses the conclusions of this work.

2 PREVIOUS WORK

Traditional tracking approaches are based on the use of models or templates that represent the target features in the spatial-temporal domain. These templates can be explicitly constructed by "hand", learned from example sequences or dynamically acquired from the moving target. These template based approaches are focused on the use of two main processes: (i) matching and (ii) updating.

Template matching corresponds to the process in which a reference template is searched for in an

Antón-Canalís L., Sánchez-Nielsen E. and Hernández-Tejera M. (2006).
SWARMTRACK: A PARTICLE SWARM APPROACH TO VISUAL TRACKING.
In *Proceedings of the First International Conference on Computer Vision Theory and Applications*, pages 221-228
DOI: 10.5220/0001372002210228

input image to determine its location and occurrence. Over the last decade, different approaches based on searching the space of transformations using a measurement similarity have been proposed for template based matching. Some of them establish point correspondences between two shapes and subsequently find a transformation that aligns these shapes. The iteration of these two steps involves the use of algorithms such as iterated closest points (ICP) (Besh et al., 1992), (Chen et al., 1992) or shape context matching (Belongie et al., 2002). However, these methods require a good initial alignment in order to converge; particularly whether the image contains a cluttered background. Other approaches, in order to compute the transformation that best matches the template into the image, are based on searching the space of transformations using exhaustive search based methods (Rucklidge, 1996). A reduction of the computational cost has been introduced by means of the use of heuristic algorithms (Sánchez-Nielsen, 2005b).

Template updating is related to the process of update of the template that represents the target. The underlying assumption behind several template tracking approaches is that the appearance of the object remains the same through the entire video (Tyng-Luh, 2004), (Comaniciu, 2000). This assumption is generally reasonable for a certain period of time and a naïve solution to this problem is updating the template every frame (Parra et al., 1999) or every *n* frames (Reynolds, 1998) with a new template extracted from the current image. However, small errors can be introduced in the location of the template each time the template is updated and this situation entails that the template gradually drifts away from the object (Matthews et al., 2004). Matthews, Ishikawa and Baker in (Matthews et al., 2004) propose a solution to this problem. However, their approach only addresses the issue related to objects whose visibility does not change while they are being tracked. An improvement of the update problem for this situation using a second order isomorphism based method has been recently proposed by (Guerra, 2005).

Other approaches based on deformable templates (Yuille et al., 1992) minimize, for each frame, an energy function which is specific to the geometry of the tracked object. Elastic snakes (Kass et al., 1987) minimize a more general energy function, which has terms representing elastic and tensile energy to ensure that the snake is smooth, and an image-dependent term that pushes the snake towards the feature of interest. The Kalman tracker (Blake et al., 1993) requires a learned linear stochastic dynamical model which describes the evolution of the contour to be tracked, assuming that the observation of the contour has been corrupted by Gaussian noise. The condensation tracker (Isard, 1998) also assumes a dynamical model describing contour motion, which requires training using the object moving over an uncluttered background to learn the motion model parameters before it can be applied to the real scene.

Currently, computing all the possible transformations that best match a template into an image and updating the new appearance of the target without drifting the tracked object for tracking arbitrary shapes with fast and vast movements under unrestricted environments for real-time tasks are open problems.

On the other hand, the main issue of deformable template based approaches is that for any given application, hand- crafting is required; that is, if it is desired to track the motion of lips, a specific energy function that is appropriate for lips must be designed. Kalman trackers solve this problem, but are not adequate to track moving objects with the presence of clutter. This problem is addressed by the condensation tracker. However, this tracker requires a dynamical model of the object to be tracked.

In this paper, a new approach is proposed to solve the problem of visual tracking of objects with arbitrary shapes in cluttered moving scenes for different visual applications under unrestricted environments. As a result, instead of using region template tracking or using salient features in the image, or minimizing energy functions, we propose to use a Swarm Intelligence metaphor based on a prey-predator scheme with a particle swarm of predators defined to track a herd of prey pixels using the intensity of its *scent*. Neither complete aspect based-templates of the visual target nor dynamical model of the motion of the object are required.

3 SWARM INTELLIGENCE

Swarm intelligence (SI) (Bonabeau, 2000) is an innovative computational and behavioral metaphor that takes its inspiration from biological examples provided by social insects and by swarming, flocking, herding and shoaling phenomena in vertebrates (Parrish et al., 1997). SI is an artificial intelligence technique based on the study of collective behaviour in decentralized, self-organized systems. SI systems are typically made up of a population of simple individuals interacting locally with one another and with their environment.

Although there is no centralized control structure dictating how individuals should behave, the main characteristic of this approach is that the collective behaviour is an emergent phenomen resulting from the interaction of the local behaviour of each independent individual. Thus, the abilities of such natural systems transcend those of individuals. The advantages of this metaphor are related, on one hand, by the robustness and sophistication of the obtained group behaviour and, on the other hand, with the simplicity and low computational costs of the individual computational elements.

Many successful SI techniques have been developed during last years, including Ant Colony Optimization (ACO), (Dorigo, 1996), or Particle Swarm Optimization (PSO) (Eberhart, 1995) as metaheuristic optimization techniques. SI simulation techniques of animal group behaviour have been used in artificial life, computer graphics and picture animation.

Among artificial life simulations, Boids (Reynolds, 1987) is an example of emergent behaviour; the complexity of Boids arises from the interaction of individual agents (boids, in this case) adhering to a set of simple rules. The rules applied in the simplest Boids world are: (1) separation, (2) alignment and (3) cohesion. This framework, related to *Steering Behaviours*, is often used in computer graphics, providing realistic-looking representations of flocks, shoals, herds or crowds.

In the following two sections, the proposed method, using the Swarm Intelligence paradigm, is described.

4 PREDATOR SWARM BASED MODEL

The tracking process is formulated in terms of a predator-prey scheme where pixels in a video sequence are considered preys and a particle swarm cooperates to hunt them.

A set of prey samples (pixels) is selected in an initial image of the video sequence. Preys are characterized by their *scent intensity*, which is an abstraction of their pixel image information: colour and gradient magnitude. In order to follow them, a swarm with the same number of predator particles is generated. Each predator particle will be fed with a single sample, and it will adapt its taste preference to that prey's features. During the video sequence, each predator will try to satisfy its taste hunting similar preys, following their scent. However, as preys may disappear due to pixel attributes changing over time, predators will be able to adapt their sense of smell in order to hunt different preys. This way, each image in the video sequence may be understood as a herd where each pixel is a potential prey for the swarm, depending on its colour and gradient value.

Predators are designed as described in the following subsections.

4.1 Swarm Structure

In order to be able to hunt its favourite preys, each particle stores the following information:

1) Position in the search space.

2) Unitary velocity in the previous iteration, initially zero.

3) Speed, the amount of pixels that a particle is able to travel between two iterations. Speed varies in time depending on a particle's comfortness (*Pcf*) (see below).

4) Colour bank list (*Pcbl*), a list of recently seen colours that is a representative subset of the colours that are similar to the colour of the presented prey pixel at initial time. Bank colours are represented by CIE L*a*b colour space. Thus, a certain light intensity independence may be obtained weighting each L*a*b colour vector when two colours (0.1*L, 1.0*a, 1.0*b) are compared.

A particle's comfortness (*Pcf*) is a measurement of its similarity with its neighbour image pixel's colour, given by:

$$Pcf = \min(sim(n,m)) \quad \forall n \in N, \forall m \in Pcbl \qquad (1)$$

Where *N* is the particle's neighbourhood, *Pcbl* is the particle's colour bank list and *sim* is a similarity measurement given by:

$$sim(n,m) = e^{\frac{|n_c - m_c|}{\sigma}} \qquad (2)$$

Where $|n_c - m_c|$ is the Euclidean distance between the two colours in CIE L*a*b* colour space. This coefficient measures the quality of prey tracking as it is carried out by the predator particle.

In order for each particle to keep contact with the swarm, three global values are computed using a weighted average of each particle's information.

1) Swarm centroid:

$$S_c = \frac{(\sum_{i<S}^{i=0} P_i \cdot Pcf_i)}{(\sum_{i<S}^{i=0} Pcf_i)} \quad (3)$$

Where P_i corresponds to the particle's position and Pcf_i represents the particle's comfortness.

2) Swarm velocity:

$$S_v = \frac{(\sum_{i<S}^{i=0} Pv_i \cdot Pcf_i)}{(\sum_{i<S}^{i=0} Pcf_i)} \quad (4)$$

Where Pv_i is the particle's velocity.

3) Predicted centroid: given the current swarm centroid and velocity, the swarm predicts where its centroid may lay in the following iteration.

Using Pcf as a weighting factor, we assure that those predator particles that are closer to their objective prey are much more relevant to the swarm's global behaviour than those particles that may have lost their target.

4.2 Swarm Behaviour

The swarm follows a Boid-like movement (Reynolds, 1987), preying those high gradient areas that best suit its particles P_{cbl} colours. Each particle follows four movement rules, each of which returns a velocity vector, where the weighted sum of them will characterize the final particle velocity and speed.

4.3 Particle Movement Rules

Swarm movement and preying behaviour emerges from the interaction of each particle's movement, which is defined by the following rules:

Rule 1) Colour & Topography: A particle analyzes its closest preys (image pixels in the neighbourhood of its initial location) obtaining a vector towards the area with higher gradient magnitude and colour similarity with the particle's colour bank.

$$V_1 = \frac{\sum_{i<N}^{i=0} (P_i - P) \cdot \min(sim(Pc_i, Pcbl)) \cdot \nabla I_i}{\sum_{i<N}^{i=0} \min(sim(Pc_i, Pcbl)) \cdot \nabla I_i} \quad (5)$$

where P_i represents a prey's position, P corresponds to the current particle's position, $sim(Pc_i, Pcbl)$ is given by expression (2) for each value stored in the particle's $Pcbl$, and ∇I_i is the gradient magnitude at pixel i. This element introduces a topographical-related weight in the equation, giving priority to significant image points (high gradient magnitude pixels) in the particle movement.

The particle's speed is computed by the following expression:

$$P_s = MINS + \frac{\sum_{i<N}^{i=0} \min(sim(Pc_i, Pcbl)) \cdot \nabla I_i}{N} \cdot MAXS \quad (6)$$

Where $MINS$ and $MAXS$ are predefined minimum and maximum speeds for a given particle. The sum is related to a measurement of how well a particle's colour fits in its neighbourhood. The higher the value (worse fitting) the faster it will move. Increasing its speed, a particle will likely escape faster that part of the image, hopefully finding better preys guided by the rest of rules.

Rule 2) Grouping: Computes a vector from the particle's position towards the current swarm centroid. This rule will avoid scattering, keeping the swarm together. A particle uses the swarm centroid instead of it closest neighbours positions like Boids do, because group splitting is not desirable. It is obtained as follow:

$$V_2 = \frac{(S_c - P)}{|S_c - P|} \quad (7)$$

Where S_c comes from (3).

Rule 3) Alignment: Computes the sum of the particle's current velocity and the swarm velocity. With this rule a particle will adapt its movement to head towards where the rest of the swarm is heading to. Once again, instead of its closest neighbours the whole swarm is considered. This rule acts like a voting system where the majority decides where the swarm will move.

$$V_3 = \frac{(S_v - P_v)}{|S_v - P_v|} \quad (8)$$

where S_v comes from (4)

Rule 4) Prediction: This rule will direct the particle's movement towards the position where the swarm's centroid will most probably be at the following iteration.

$$V_4 = \frac{(S_{pc} - P)}{|S_{pc} - P|} \quad (9)$$

Where S_{pc} corresponds to the swarm predicted centroid position. This way, a particle is able to guess the group position in future iterations.

The classic Boids separation behaviour (Reynolds, 1987) was not included in our swarm model because each particle has its own colour information, its own prey, so even if two particles share the same spatial position they do not have to necessarily move towards the same point.

Finally, the four resultant velocities are weighted and added to a portion of the previous iteration velocity for each particle, Ps_{t-1} and multiplied by the current particle speed.

$$V = (v_1.w_1 + v_2.w_2 + v_3.w_3 + v_4.w_4 + Ps_{t-1}).Ps_t \quad (10)$$

5 TRACKING

The cooperative social interaction leads the swarm towards those areas in the image which are similar to that where the swarm was created, emerging a non-structural pattern tracking behaviour where the swarm centroid, velocity and speed will respectively indicate the tracked object position and relative movement information, as seen in figure 1.

Figure 1: Object being tracked (preyed) by a swarm. White dots represent particles, while the white line shows current swarm velocity.

Tracking is enhanced using two key ideas: (i) individual comfortness optimization and (ii) swarm adaptation.

Individual comfortness optimization is related to a direct application of Particle Swarm optimization theory (Eberhardt, 1995); where each particle tries to minimize a certain error using local and global information based on colour matching and gradient's magnitude. As a result, each particle will move towards those prey pixels that best match the tracked *scent* (colour). Note that prey pixel *scent* intensities are proportional to image gradient magnitudes, so predators will be attracted to interest points in images that match their *scent* track.

Figure 2 shows a detail of those points that seem to be most interesting to a swarm that is tracking a white road line in an automated vehicle based context.

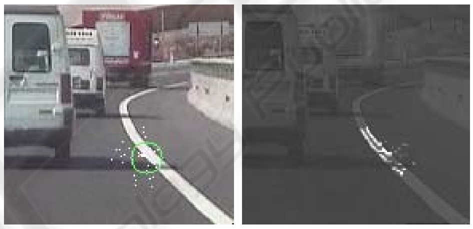

Figure 2: Swarm perception. A swarm created in a white region will be attracted by white colours on high gradient magnitude pixels, shown brighter on the image on the right.

In order to avoid the introduction of small errors in the location of the swarm, the swarm is updated using a colour bank for each particle. This colour bank will allocate a list of similar prey scents, avoiding any kind of false averaged values when a particle is comparing itself with its neighbourhood as seen in section 4.1, using rule 1.

6 RESULTS

In order to test the proposed tracking approach, different indoor and outdoor video streams related to different visual tasks have been used for experimental evaluations. Each one of these sequences contains frames of 320 x 200 pixels that were acquired at 25 fps. All experimental results were computed on a P-IV 1.4 Ghz.

Prey samples are initialized defining a rectangular area on the first image of the sequence. This process can be automated, e.g. using cascade classifiers for face or hand detection (Anonymous). The swarm, once created and fed with sample prey pixels, is able to follow them on a varied number of non-cluttered backgrounds and light conditions, as seen in Figure 3, 4, 5, 6 and 7.

The parameters used for the swarm (100 particles) have been initialized with the following values: W_1 (colour & topography) = 1.0, W_2 (grouping) = 0.3, W_3 (alignment) = 0.5, W_4 (prediction) = 0.2, W_5 (Particle's velocity at time t-1) = 0.1, δ= 10.0, neighbourhood size = 15, minspeed = 5.0, maxspeed = 10.0 and colour bank list size = 3.

The achieved processing rate is around 15fps. Note no optimizations have been implemented.

In order to evaluate the robustness of the proposed approach, we manually annotate the centroid point to be tracked and then we measure the Euclidean distance from the annotated hand-tracked point and the swarm's centroid to the origin (0, 0) through time. Values were measured every ten frames. Graphics in figure 3, 4, 5, 6 and 7 illustrate the results obtained.

The dotted line represents the hand-tracked point and the continuous line corresponds to the swarm's centroid. It is important to point out that the swarm floats freely over tracked objects, so both lines will not necessarily coincide. However, they evolve similarly when the swarm follows successfully the tracked object.

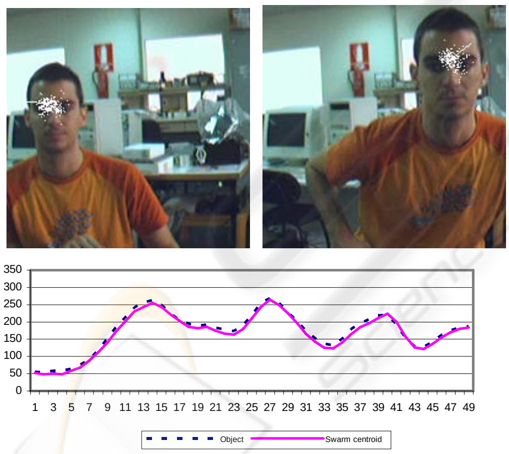

Figure 3: The swarm is created over a face, and follows it while it moves around.

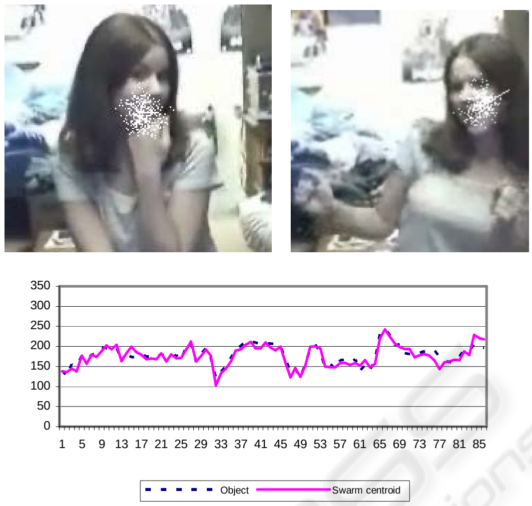

Figure 4: The swarm is created over a girl's face, and follows it while she makes faces and moves around. The swarm loses its target when it is hidden almost at the end of the sequence.

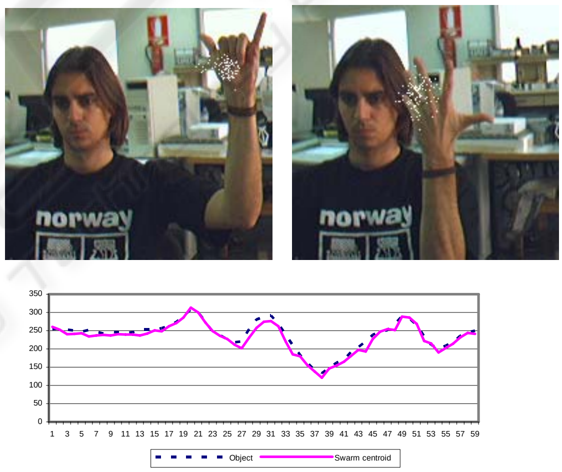

Figure 5: This time, our swarm follows a continuously gesture changing hand. It has no problems even when the hand meets the face on its movement.

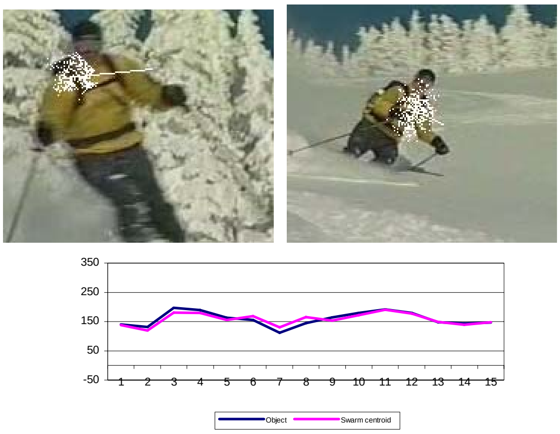

Figure 6: The swarm follows a skier, who moves in a fast wavy course. Sudden changes in speed (acceleration) and direction confuses the swarm, but it is able to follow the skier.

Figure 7: A car is followed by the swarm while it drives away, until it becomes too small for the swarm to follow it.

On Figure 7 the car is lost when the shape that characterizes the car is too small and the colour and gradient magnitude are not significant for the swarm. Background areas with high magnitude gradient and significant colours for the swarm may also attract it.

A swarm, however, may deal with occlusion as long as it has tracked an object for some frames and it does not alter its movement during occlusion. When this happens, rule 1's resultant vector will not be significant. The swarm's acquired velocity (rule 3) will allow it to surpass the occlusion. However, if the occluding object's features satisfy the swarm's taste, it may decide to follow it and lose its original target. In general, swarms may be confused by those areas with high gradient magnitudes and colours similar to what the swarm expects. This could be solved creating leaders in the swarm, able to follow feature points in the tracked object, which would have a higher influence over the swarm's movement.

The amount of weights and parameters could be seen as a drawback of the proposed method. However, once a good set of parameters have been computed, the proposal works for a wide range of visual applications and arbitrary shape with a vast range of movements such those illustrated in Figure 3, 4, 5, 6 and 7.

7 CONCLUSIONS AND FUTURE WORK

In this paper, a new tracking method based on a Swarm Intelligence Metaphor has been described. The main idea of the proposal consists on a prey-predator scheme, where a swarm of predator particles follows pixel *scents* (colours) similar to those that where presented to predators at initial tracking time. Image gradient is used as a feature regulator, defining the *scent intensity*, which is proportional to the value of the gradient. Thus, matching colours located in high interest pixels are much more interesting for a given predator. Each predator particle's movement is governed by four basic rules. Tracking behaviour emerges from the interaction of each particle, where the tracked object's position is defined by the swarms' centroid.

Because our swarms do not follow shapes but light intensity independent colours, the resulting tracking method is robust under deformations of the tracked object, cluttered images and ligh changes, being computationally a low cost solution.

Experimental results show that, with unrestricted images, and using general purpose hardware, almost real time tracking is obtained (~20fps, tracking with 100 particles, using 320*200 pixels images in a P-IV 1.4 Ghz). Due to its computational simplicity the proposed solution is very efficient and highly parallelizable.

The method's accuracy is based on the size of the tracked object. With a good area to track, as e.g, sequences in Figures 3 and 4, accuracy is maximum, decreasing proportionally to the size of the region to be tracked, such as in the last frames of the sequence corresponding to Figure 7. Future work will include comparisons with classic tracking methods.

ACKNOWLEDGMENTS

This work has been supported by the Spanish Government and the Canary Islands Autonomous Government under projects TIN2004-07087 and PI2003/165.

REFERENCES

Aloimonos, Y., 1993. Active Perception. Lawrence Erlbaum Assoc., Pub., N.J.

Belongie, S., Malik, J., and Puzicha J., 2002. Shape matching and object recognition using shape context. In IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(4):509-522.

Besh, P. J., and McKay N., 1992. A method for registration of 3D shapes. In IEEE Trans. on Pattern Analysis and Machine Intelligence, 14(2):239-256.

Blake, A., Curwen R., and Zisserman A., 1993. A framework for spatio-temporal control in the tracking of visual contours. In International Journal of Computer Vision, 11(2):127-145.

Bonabeau, E., Dorigo, M., and Theraulaz, G., 2000. Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press.

Chen Y. amd Medioni G., 1992. Object modelling by registration of multiple range images. In Image and Vision Computing, 10(3):145-155.

Comaniciu, D., Ramesh V., and Meer P., 2000. Real-time tracking of non-rigid objects using mean shift. In IEEE Conf. on Computer Vision and Pattern Recognition, volume II, pp. 142-149, Hilton Head, SC.

Dorigo M., V. Maniezzo & A. Colorni, 1996. Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1):29-41.

Eberhart, R. C. and Kennedy, J., 1995. A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan. pp. 39-43.

Guerra Cayetano, Hernández Mario, Domínguez Antonio, Hernández Daniel, 2005. A new approach to the template update problem. In Lecture Notes in Computer Science LNCS 3522, pp. 217-224.

Isard, M., and Blake A., 1998. Condensation-conditional density propagation for visual tracking. In International Journal of Computer Vision, 29(1):5-28.

Kass, M., Witkin A., and Terzopoulos D. Snakes: active contour models. In Proc. 1st International Conference on Computer Vision.

Kennedy, J. and Eberhart, R. C., 1995. Particle swarm optimization. In Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948.

Matthews, I., Ishikawa, T., and Baker S., 2004. The template update problem. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(6):810-815.

Parra, C., Murrieta-Cid, Devy, M., and Briot, M., 1999. 3-D Modelling and Robot Localization from Visual and Range Data in Natural Scenes. In Lecture Notes in Computer Science 1542, Springer Verlag, pp. 450-468.

Parrish, Julia K. (Editor), William M. Hamner (Editor), 1997. Animal Groups in Three Dimensions: How Species Aggregate. Cambridge University Press.

Pentland, A., 2000. Perceptual Intelligence. In Communications of ACM, 43(3):35-44.

Reynolds, C. W., 1987. Flocks, Herds, and Schools: A Distributed Behavioural Model. In Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings) pp. 25-34.

Reynolds, J., 1998. Autonomous underwater vehicle: vision system. PhD thesis, Robotic Systems Lab. Department of Engineering. Australian National University Canberra, Australia.

Rucklidge W. J., 1996. Efficient Visual Recognition Using the Hausdorff Distance. In Lecture Notes in Computer Science, nº 1173, Springer-Verlag, NY.

Sánchez-Nielsen Elena, Hernández-Tejera Mario, 2005a. A fast and accurate tracking approach for automated visual surveillance. In 39th IEEE International Carnahan Conference on Security Technology, pp. 113-116.

Sánchez-Nielsen Elena, Hernández-Tejera Mario, 2005b. A heuristic search based approach for moving objects tracking. In 19th International Joint Conference on Artificial Intelligence (IJCAI-05), pp. 1736-1737.

Turk, M., 2004. Computer Vision in the Interface. In Communications of the ACM, 47(1): 61-67.

Tyng-Luh Liu, Hwan-Tzong Chen, 2004. Real-Time tracking using trust-region methods. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(3):397-401.

Yuille, A., Hallinan, P., and Cohen D., 1992. Feature extraction from faces using deformable templates. In International Journal of Computer Vision 8(2):99-112.