Dynamic Open Contours Using Particle Swarm Optimization with Application to Fluid Interface Extraction

M. Thomas[1], S.K. Misra[2], C. Kambhamettu[1], and J.T. Kirby[2]

[1] Video/Image Modeling and Synthesis Lab, Dept. of Computer and Info. Sciences,
University of Delaware, Newark, DE

[2] Center for Applied Coastal Research, Dept. of Civil and Env. Engineering,
University of Delaware, Newark, DE

Abstract. This paper describes a method for the estimation of a dynamic open contour by incorporating a modified particle swarm optimization technique. This scheme has been applied to a "Particle Image Velocimetry" experiment for the analysis of fluid turbulence during a hydraulic jump. Due to inter reflections within the medium and refractions across different media interfaces, the imagery contains spurious regions, which have to be eliminated prior to the estimation of turbulence statistics at the fluid surface. The PIV image sequences provide a strict test bed for the performance analysis of this estimation mechanism due to the occurrence of intense specularity and extreme non-rigid motion dynamics.

1 Introduction

Edge detection and image segmentation is a crucial initial step in most computer vision applications prior to performing high-level tasks such as object recognition and scene interpretation. The presence of noise and other non-linearities imposes a strict restriction on this segmentation process. Since its formulation, the active contour model [1] tries to combine low level image information with high level structural information to provide a lucid description of the underlying structure in the presence of non-linearities. Usually this balance is brought about by two energy components, an internal energy component that characterizes the contour smoothness making it possible to estimate contour elements in places with incomplete image information and an external energy component that incorporates the low level image characteristics.

Among the variants of the active contour, notable ones include the greedy algorithm proposed by Williams and Shah [2], the balloon model by Cohen [3], the region based model by Ronfard [4] and the gradient vector flow based snake formulations by Xu and Prince [5]. Contour modeling via state space estimation was performed by Isard et al. [6] where the contour was represented as a state element and sequential importance sampling was used to track the contour state over time. Pérez et al. [7] described a contour extraction procedure,

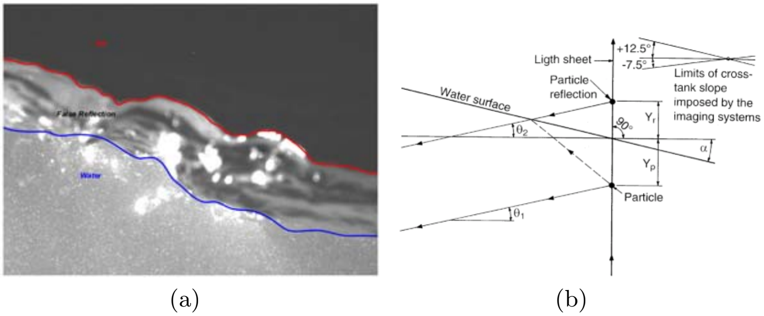
P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3851, pp. 643–652, 2006.

called Jetstream, that was also based on importance sampling with each contour location being used to compute the position of next contour location. Most active contour formulations depend on the availability of high image gradient for efficient processing. In image sequences with weak gradient information, these methods have difficulty in estimating the contour accurately. Statistical snakes as proposed by Ivins et al. [8] and the discriminant snakes proposed by Pardo et al. [9] tackle contour formulation by incorporating statistical information from the image and thus have been shown to be robust under noise and low gradient imagery.

In this paper, we attempt to extract a dynamic open contour that is built along the lines of the statistical snakes with multiple candidate hypotheses extracted from the image via a modified swarm optimization model. The swarm optimization scheme, "consume and move" has been developed to obtain multiple candidates, which could subsequently be used in computing the contour. The paper begins by giving a brief description of the interface extraction problem. This is followed by the description of the Particle Swarm Optimization model. The processing methodology that was developed for the minimization framework is described subsequently along with the results obtained from the algorithm. Finally, we present our conclusions and possible future directions.

2 Problem Description

In analyzing the salient structures in the velocity fields of incompressible turbulent fluid flows, such as water in confined channels [11], insertion of probes and measuring gauges into the fluid flow could create artificial turbulent deformations. In a regular Particle Image Velocimetry (PIV) experiment, the flow is seeded with suitable tracer particles, illuminated by a planar laser sheet and time-lapsed images are recorded. The displacement of the particles in the images is measured in the plane of the image, and is used to determine the flow (see [12] and the references therein). PIV has thus become an established non-



(a) (b)

Fig. 1. (a) Example of the interface extraction problem (b) Geometry of the problem - *Image taken from [10], Figure 4, page 432* intrusive measurement technique to measure the kinematics of turbulent fluid flow in controlled laboratory experiments.

It is often imperative to obtain detailed instantaneous flow velocities near the air-water (2-phase) interface, which necessitates an accurate estimation of the interface. This is inherently a difficult problem since most intensity based edge detection methods fail due to the presence of interface reflections in a PIV image [13]. Typically, the interface is concurrently visualized by a technique called Laser Induced Fluorescence (LIF) in which a fluorescent dye is added to one phase and excited to a particular wavelength by the laser thereby obtaining the interface as a sharp gradient at the specific wavelength[14]. A simultaneous PIV and LIF experiment therefore requires two separate imaging systems which add both complexity and cost to the entire estimation process.

Given the characteristics of fluid flow, the main problem that arises in estimating the interface from cross sectional images are the presence of badly defined boundaries that occurs due to the translucency of the fluid. The other problem that is often encountered is the presence of false regions of reflection (Fig. 1(a)). These regions occur due to the imaging device, which captures light undergoing total internal reflections from various sections in the fluid flow (Fig. 1(b)). Manual calculation of the interfaces remains a daunting task due to the large volume of data that is typically obtained in a regular PIV experiment. A robust, objective and automated method, which would be able to tackle these problems and calculate the interface solely based on the available image information, is thus very essential.

3 Processing Methodology

3.1 Particle Swarm Optimization

"Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique for optimizing complex functions through the *interaction of individuals* in a population of particles." ([15], pp 2). The original formulation was proposed by Kennedy and Eberhart [16] and was based on the simulation of social behavior among flocks of birds. Each particle in the population (also called the swarm) adjusts its trajectory towards its own best position and towards the best position attained by the whole group [17]. The system dynamics are governed by the following equations.

$$\mathbf{v}_i^{(t)} = \omega \mathbf{v}_i^{(t-1)} + c_1 \chi_1 (\mathbf{p}_i^{(t-1)} - \mathbf{x}_i^{(t-1)}) + c_2 \chi_2 (\mathbf{g}^{(t-1)} - \mathbf{x}_i^{(t-1)}) \tag{1}$$

$$\mathbf{x}_i^{(t)} = \mathbf{x}_i^{(t-1)} + \mathbf{v}_i^{(t)} \tag{2}$$

where $\chi_1, \chi_2 \sim U[0,1]$ are two $N_s \times N_s$ diagonal matrices of uniform random numbers with N_s being the total number of particles in the swarm. ω is the "inertia weight" that regulates the trade-off between the global (wide-ranging) and the local (nearby) exploratory capabilities of the swarm [17]. $\mathbf{x}_i^{(t-1)}$ is the i^{th} particle in the swarm at the $(t-1)^{th}$ iteration and $\mathbf{v}_i^{(t-1)}$ is its corresponding "velocity" component. $\mathbf{p}_i^{(t-1)}$ corresponds to the position of the best fitness value for the i^{th} particle while $\mathbf{g}^{(t-1)}$ corresponds to the best fitness value for the entire swarm.

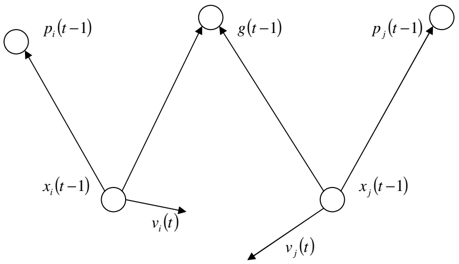


Fig. 2. Particle system in the Particle Swarm Optimization model for a two particle ($\mathbf{x}_i^{(t-1)}$ and $\mathbf{x}_j^{(t-1)}$) system

Among the three components of this dynamical equation, $\omega\mathbf{v}_i^{(t-1)}$ is the "inertial component", which constrains the velocity state estimate along the direction of $\mathbf{v}_i^{(t-1)}$. The second component, the "cognitive term" for each particle, $c_1\chi_1(\mathbf{p}_i^{(t-1)} - \mathbf{x}_i^{(t-1)})$ constrains the particle motion in the direction of its previous best value while the third component, the "social component", $c_2\chi_2(\mathbf{g}^{(t-1)} - \mathbf{x}_i^{(t-1)})$, directs the particles towards the best among all the elements in the swarm. The random variables χ_1 and χ_2 provide for the stochastic parameters for the search with c_1 and c_2 as two positive weights that control each of the components (Figure 2). An important aspect of PSO systems, for performing functional optimizations, is that the entire dynamical update is performed using additions and multiplications alone and is thus computationally very efficient.

"Explorers and Settlers" Paradigm. One of the variants to the particle swarm model was the "Explorers and Settlers" model as proposed by Kennedy and Eberhart [16]. In this paradigm, the swarm is composed of having two kinds of agents, the "explorers" and the "settlers". The "settlers" provided for micro-level function optimization of "known" regions of the problem domain while the "explorers" searched for regions outside for better "solutions". But as discussed by Kennedy and Eberhart [16] this scheme did not provide a significant improvement in the tests that they conducted.

In contrast, tackling the interface problem requires estimating multiple candidate hypothesis from a given search space so that the final open contour is drawn across the best possible candidates. To tackle this requirement we modified the "explorers and settlers" scheme to provide a mechanism for the swarm to continue in the exploratory phase after a goal is reached. This model, called the "consume and move", can be described in terms of migratory systems where the swarms continue moving in search for new pastures after the consumption of one specific region.

The essential principle of this model is to decrease the fitness metric of the search space after the swarm has converged at a specific goal $\mathbf{g}^{(t)}$. Depending on ω, c_1 and c_2, a subset of particles $x_i^{(t)}, i = \{1 \ldots N_1\}$ $(N_1 \leq N_s)$, would converge to the best fitness value in the search space. For every particle $\mathbf{x}_i$ in the swarm, the fitness functional is cumulatively scaled down in proportion to the proximity of the particle to the goal. This scaling could be accomplished using the affinity function $\exp(-\|\mathbf{x}_i^{(t)} - \mathbf{g}^{(t)}\|^2/2\sigma^2)$ with the fitness at the position of the particles that coincide with the goal node being scaled more than the others. This scaling would thus "consume" the fitness functional to a greater extent at regions that are at a closer proximity to point of convergence of the swarm particles. Thus, iterating the search mechanism with this modified fitness space would constrain the swarm to "move" out and look for other possible candidate positions.

3.2 Contour Estimation

Kass [1] defined an active contour as a parametric contour $\mathbf{v}(s) = (x(s), y(s)), s \in [0, 1]$ that balances the internal energies E_{int} and the external energies E_{ext} (Eq. 3)

$$E^* = \int_0^1 [w_1 E_{int}(\mathbf{v}(s)) + w_2 E_{ext}(\mathbf{v}(s))] ds \tag{3}$$

where w_1 and w_2 are the weights that control the importance of one energy term over the other. Assuming a discrete approximation of Eq. 3, we have

$$E = \sum_{i=1}^{N} [\alpha E_{dist}(\mathbf{v}_i) + \beta E_{smo}(\mathbf{v}_i) + \gamma E_{ext}(\mathbf{v}_i)] \tag{4}$$

where α, β, γ are the weighting parameters, N is the number of discrete contour samples and

$$E_{dist}(\mathbf{v}_i) = |\frac{(\|\mathbf{v}_i - \mathbf{v}_{i-1}\| + \|\mathbf{v}_{i+1} - \mathbf{v}_i\|)}{\frac{2}{N-1} \Sigma_{j=2}^{N} \|\mathbf{v}_j - \mathbf{v}_{j-1}\|} - 1|$$

$$E_{smo}(\mathbf{v}_i) = 1 - \cos(\theta_i) = 1 - \frac{(\mathbf{v}_{i+1} - \mathbf{v}_i) \cdot (\mathbf{v}_i - \mathbf{v}_{i-1})}{\|\mathbf{v}_{i+1} - \mathbf{v}_i\| \|\mathbf{v}_i - \mathbf{v}_{i-1}\|}$$

with θ_i being the smoothness term as defined in [18]. The external energy E_{ext} is derived from the image information and is usually the magnitude of the image gradient information.

Pixel Likelihood Estimation. The external energy term (E_{ext}) in a dynamic contour transfers the low level image information to the high level structural information. Most active contour methods are derived using the image gradient as the external energy constraint, but in images where the gradient is hard to estimate or the estimated gradient is inaccurate, the external energy functional has to be modeled using other image characteristics.

In a PIV image, the tracer particles have a distinct signature [19] which would enable a high-pass filter to approximately intensify the particle zones and

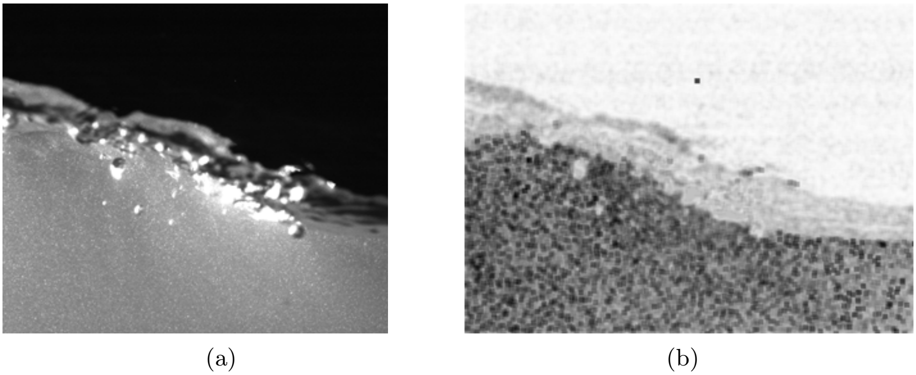


Fig. 3. (a) Input PIV image (b) PIV image after PCA over the feature space

suppress the other regions. Local image statistics such as entropy, mean and standard deviation are extracted from the high-pass filtered image $I_{\mathcal{H}}$. Each of these features would provide information regarding the local data variation in $I_{\mathcal{H}}$. Principal Component Analysis (PCA) of this feature space is subsequently used to transform this space so as to maximize the data variation along individual eigen-directions. As shown in the figure 3(b), this PCA based transformation of the local image statistics provide a more succinct description of the underlying particle distribution.

In the PIV data, the likelihood that a pixel belongs to the surface was computed as the difference between the average intensity on the top and bottom of the pixel spatial position. Pixels very close to the actual surface, described a higher value of the likelihood as against the pixels elsewhere. This likelihood formed the fitness metric that was used in the "consume and move" swarm optimization strategy (section 3.1) to obtain multiple snaxel candidates.

Contour Optimization. Given the candidate hypotheses, finding the contour that minimizes the internal and external energy functional is performed using the dynamic programming (DP) approach as described by Amini et al. [20]. Depending on the pixel likelihood, different snaxels would have different hypotheses and thus energy minimization using DP is well-suited in tackling this contour optimization problem. The overall energy minimization would be achieved by minimizing the intermediate variables ξ_k such that

$$\mathbf{v}_k = \min \xi_k,\ k = 1 \dots N \tag{5}$$

under the constraint that

$$\xi_k = \xi_{k-1} + \min_{\mathbf{z}_k \in \mathcal{C}_k} \{E_{int}(\mathbf{z}_{k-1}, \mathbf{z}_k, \mathbf{z}_{k+1}) + E_{ext}(\mathbf{z}_k)\} \tag{6}$$

where $\mathbf{z}_k$ are the candidate snaxels positions and E_{int} and E_{ext} are the internal and external energies computed at each of the candidate positions $\in \mathcal{C}_k$. ξ_1 is initialized as the $\min_{\mathbf{z}_1 \in \mathcal{C}_1} E_{ext}(\mathbf{z}_1)$. The completion of the entire forward and reverse iterations of the DP presents the best possible positional estimates, $\mathbf{v}_k$.

4 Experimental Setup

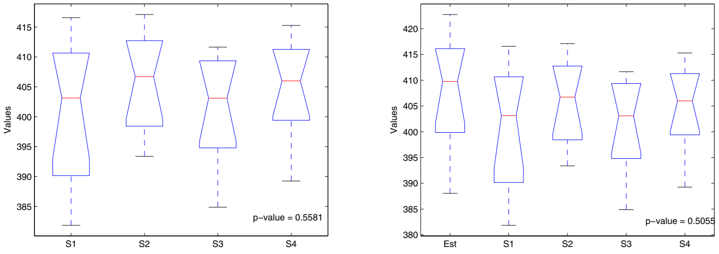
The experiments were performed in a recirculating water tank that is sixteen feet long and one foot wide, with glass side walls and a solid bottom. The water was seeded with 14 μm silver coated hollow glass spheres and was pumped into the upstream end of the channel. A 120 mJ/pulse Nd-Yag New Wave solo laser source was mounted onto a custom-built submersible periscope which was lowered into the water so that the laser beam emerged as a planar light sheet parallel to the water tank wall. The flow was captured by a Kodak Megaplus 1.0 camera with a 1016 × 1008 pixel resolution (see [19] for details).

5 Results and Analysis

The algorithm that has been developed in the previous section has been tested on 1020 Particle Image Velocimetry image pairs. As can be easily seen these images are subject to extreme non rigid motion due to the fluid motion being captured and would thus be ideal in testing out the efficiency of the algorithm. The current implementation of the interface calculation is embedded in a hierarchical framework with coarse initial contours being used to guide subsequent finer contours. The entire process is iterated until the cumulative temporal variation of the contour elements, $\|\mathbf{v}_{1:N}(t-1) - \mathbf{v}_{1:N}(t)\|$ is $\leq 0.1N$, which is used as a metric to indicate the stabilization of the contour.

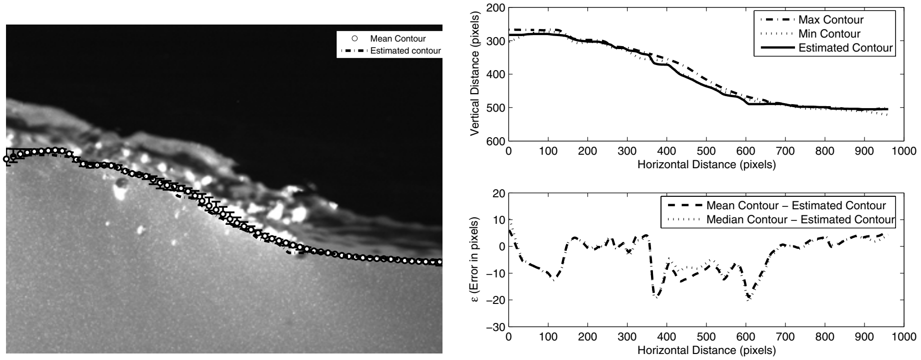
5.1 Quantitative Comparison

Due to the unavailability of ground truth, the result from the algorithm was assessed with respect to human perception. 10 randomly sampled PIV images were distributed among 4 participants with expertise in fluid dynamics. A short problem description was provided and the subjects were asked to find out the contour as they best perceived it. Since the inputs obtained from the subjects were sparse, a least square B-spline was used to compute the contour for the entire width of the image. In comparing the results of the algorithm with the output from the human participants, it is essential that the contours computed by the participants be considered as **NOT** significantly different from one another. It is also necessary to statistically show that the estimated contour does not significantly differ from those obtained from the participants. This analysis was accomplished using an independent sample one-way ANOVA. Figure 4(a) shows the box plots for the average contour variation across the 10 images for each of the participants (S1, S2, S3, S4). Figure 4(b) shows the average contour variation of the estimated contour in tandem with the output from the four participants. Also indicated are the corresponding p-values to provide a quantitative metric of similarity between the contours across the image pairs. The



(a) (b)

Fig. 4. Results using the One-way ANOVA (a) Testing for inter subjects variability (p-value = 0.5581) (b) Testing if the estimated contour differed significantly from the “ground truth” contours (p-value = 0.5055)



(a) (b)

Fig. 5. (a) Estimated contour points plotted in comparison with the mean variation of the “ground truth” contours (b) Mean and Median error deviation of the estimated contour

large p-values indicate that the individual sample means are not significantly different from one another or from the contour estimated by our algorithm.

Figure 5 is one of the test images (Contour 68) shown in conjunction with the error between the estimated contour and the mean of the contours from the 4 participants. The error bars show the maximum deviation from the mean and it is evident that the estimated contour falls within the upper and lower bounds of the contours obtained from the experts to a large extent. This method, thus provides a good initial estimate so as to apply free-surface kinematics to determine the exact position of the interface.

The algorithm was developed using MATLAB and has been tested with 1020 PIV images. Repeated trials indicate that the algorithm is stable and computationally efficient (The algorithm required ∼ 20 seconds to process a 128 × 960

Table 1. Error Analysis for 10 images, where the ground truth was extracted by the 4 subjects with expertise in fluid dynamics. The plot indicates the error variation ($\frac{1}{N}\sum_i^N \parallel \mathbf{x}_i - \mu_i \parallel$) between the contour estimated by the algorithm and the ground truth.

Contour	Subject$_1$	Subject$_2$	Subject$_3$	Subject$_4$
4	0.6527	0.6245	0.7504	0.7682
8	0.2791	0.2389	0.3310	0.3008
14	0.3493	0.3343	0.6700	0.6099
29	0.3550	0.5628	0.5312	0.5964
33	0.3687	0.2692	0.3573	0.4072
68	0.3998	0.2193	0.2364	0.2398
269	0.3503	0.4056	0.3093	0.3663
217	0.6549	0.3696	0.3262	0.3151
301	0.6733	0.2508	0.3732	0.4026
362	0.3813	0.3540	0.3454	0.3488

image, using $\sim$ 100 particles at each contour location). The average MSE, across the 10 images, are shown in Table 1 to further clarify the accuracy of the algorithm.

6 Conclusions

This paper describes a method to extract dynamic contours using particle swarm optimization and dynamic programming. The algorithm is robust and computationally efficient. The algorithm developed was applied to the free surface estimation in a 2-phase fluid flow using a PIV setup. Due to the lack of ground truth, the estimated contours have been compared with the results obtained from experts in the field of fluid dynamics. It has been observed that the estimated contour is not statistically different from the expert estimation. The method has now been tested over a sequence of 1020 PIV image pairs that have been further processed to compute instantaneous and ensemble average velocities at the interface.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. IJCV **1** (1988) 321–331
2. Williams, D.J., Shah, M.: A fast algorithm for active contours and curvature estimation. CVGIP: Image Underst. **55** (1992) 14–26
3. Cohen, L.D.: On active contour models and balloons. CVGIP **53** (1991) 211–218
4. Ronfard, R.: Region based strategies for active contour models. International Journal of Computer Vision **13** (1994) 229–251
5. Xu, C., Prince, J.L.: Snakes, shapes, and gradient vector flow. IEEE Transactions on Image Processing **3** (March 1998) 359–369

6. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. International Journal of Computer Vision **29(1)** (1998) 5–28
7. Pérez, P., Blake, A., Gangnet, M.: Jetstream: Probabilistic contour extraction with particles. In: ICCV. (2001) 524–531
8. Ivins, J., Porrill, J.: Statistical snakes: Active region models. In: Proc. British Machine Vision Conference. Volume 2. (1994) 377–386
9. Pardo, X.M., Radeva, P.: Discriminant snakes for 3d reconstruction in medical images. In: Proc. International Conference on Pattern Recognition (ICPR'00). Volume 4. (2000)
10. Peirson, W.L.: Measurement of surface velocities and shear at a wavy air-water interface using particle image velocimetry. Experiments in Fluids **23** (1997) 427–437
11. Zhong, J., Huang, T.S., Adrian, R.J.: Extracting 3d vortices in turbulent fluid flow. IEEE Transactions on Pattern Analysis and Machine Intelligence **20(2)** (1998) 193–199
12. Raffel, M., Willert, C., Kompenhas, J.: Particle Image Velocimetry, a Practical guide. Springer-Berlin (1998)
13. Lennon, J.M., Hill, D.F.: Particle image velocimetry measurements of undular and hydraulic jumps. Submitted to J. Hydraulic Engg. (2004)
14. Westerweel, J., Hofmann, T., Fukushima, C., Hunt, J.: Experimental investigation of the turbulent/non-turbulent interface at the outer boundary of a self-similar turbulent jet. Experiments in Fluids **33** (2002) 873–878
15. Battiti, R., Brunato, M., Pasupuleti, S.: Do not be afraid of local minima: Affine shaker and particle swarm. Technical report DIT-05-049, university of Trento, Italy (2005)
16. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks. (1995) 1942–1948
17. Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization method for constrained optimization problems. In: Proceedings of the Euro-International Symposium on Computational Intelligence. (2002)
18. Akgul, Y.S., Kambhamettu, C.: A coarse-to-fine deformable contour optimization framework. IEEE Transactions on Pattern Analysis and Machine Intelligence **25(2)** (2003) 174–186
19. Misra, S.K., Thomas, M., Kambhamettu, C., Kirby, J.T., Veron, F., Brocchini, M.: Estimation of complex air-water interfaces from PI̧V images. Experiments in Fluids (accepted for publication)
20. Amini, A., Weymouth, T., Jain, R.: Using dynamic programming for solving variational problems in vision. PAMI **12** (1990) 855–867