LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-TR-201054

Autonomous Motion Segmentation of Multiple Objects in Low Resolution Video Using Variational Level Sets

M. Moelich

November 19, 2003

AUTONOMOUS MOTION SEGMENTATION OF MULTIPLE OBJECTS IN LOW RESOLUTION VIDEO USING VARIATIONAL LEVEL SETS

Mark Moelich
Mathematics Department, UCLA
405 Hilgard Avenue, Los Angeles, CA 90095
mmoelich@math.ucla.edu

October 15, 2003

Abstract This report documents research that was done during a ten week internship in the Sapphire research group at the Lawrence Livermore National Laboratory during the Summer of 2003. The goal of the study was to develop an algorithm that is capable of isolating (segmenting) moving objects in low resolution video sequences. This capability is currently being developed by the Sapphire research group as the first stage in a longer term video data mining project. This report gives a chronological account of what ideas were tried in developing the algorithm and what was learned from each attempt. The final version of the algorithm, which is described in detail, gives good results and is fast.

Keywords - Segmentation, tracking, video, level sets, surveillance, data mining.

1 Introduction

My doctoral research at UCLA has focused on variational and stochastic methods for tracking objects in video sequences. During the Summer of 2003, I worked in the Sapphire research group (led by Chandrika Kamath), which is part of CASC at the Lawrence Livermore National Laboratory. The Sapphire group is involved with several aspects of data mining and have recently become interested in extracting, or "mining", information from very long video sequences. They are interested in understanding the behavior of multiple objects over time from video obtained, for example, from a surveillance camera placed in a public area. The first step in this long-term study is to develop algorithms that are able to isolate and track the objects of interest (e.g., people and cars) in the scene. They are also considering a fixed camera at this stage of the study. Chandrika Kamath is directing the group to develop algorithms that are able to extract the desired data from reduced resolution (in both space and time) video. Since very long video sequences require large amounts of storage, even when compressed, there is a desire to have an algorithm that can extract as much relevant information from as little actual video data as possible. The consideration of low-resolution video seems natural, yet it does not appear much in the literature, and seems to have been motivated by the lab's experience handling very large data sets. This consideration requires that new algorithms be developed since most of the standard ones assume small frame-to-frame motion. My objective for the summer was to develop an algorithm that is capable of segmenting moving objects in fixed camera, low resolution video sequences.

Based on my research at UCLA, I used a region-based, variational level set method [3, 1] to solve this problem. The resulting algorithm is an energy (or functional) that depends upon the content of the video sequence and whose minimum gives the desired segmentation. In contrast to traditional image processing techniques, this approach has a strong mathematical basis and unlike local methods, such as optical flow estimation, is effective at low frame rates.

2 Chan-Vese-based Tracking Algorithm

As a first step, I considered using the Chan-Vese-based tracking algorithm [3], which I developed in 2002. This algorithm was designed to track a single object in the presence of camera motion. The algorithm tracks an object by sequentially segmenting the frames of a video using a modified version of the Chan-Vese algorithm [1]. The final segmentation of one video frame is used as the initial contour for the next, and the initial contour for the first frame is supplied by the user of the program. The segmentation is done by minimizing an energy of the form

$$E(C;\, c_{in}, c_{ext}) = \lambda_{int} \int_{\mathrm{int}(C)} (I(x) - c_{in})^2\, dx + \lambda_{ext} \int_{\mathrm{ext}(C)} (I(x) - c_{ext})^2\, dx \qquad (1)$$
$$+ \mu\, \mathrm{length}(C) + \nu\, \mathrm{area}(\mathrm{int}\,(C)),$$

where I is the current image frame, $\mathrm{int}(C)$ and $\mathrm{ext}(C)$ are the regions interior and exterior to the segmentation contour C; c_{in} and c_{ext} are the average intensities of the interior and exterior regions, respectively; and λ_{int}, λ_{ext}, μ, and ν are design parameters.

I tested this algorithm on several of the cars in one of the test sequences [6]. Figure 1 illustrates the segmentation of a single video frame, where (a) shows the initial contour, (b) shows an intermediate contour, and (c) shows the final segmentation.

(a) (b) (c)

Figure 1: Evolution of the Chan-Vese segmentation contour: initial contour (a), an intermediate contour (b), and final contour (c).

Figure 2 shows the result of the tracking algorithm over time. In this example, the algorithm was applied to each frame of video and the results shown at approximately one second intervals.

(a) (b) (c)

Figure 2: Results of the Chan-Vese-based tracking algorithm over time. Each frame of video was used in tracking the object and the results are shown at approximately one second intervals.

Although the above results were generated by processing each frame of video, several frames of video can be skipped. The only restriction is that the final contour of one iteration has some overlap with the object in the next frame. Although the algorithm is effective at tracking a single object, it is not practical for multiple objects because each object requires a separate level set function and segmentation. Also, it is not a good candidate for autonomous tracking since it requires the user of the program to supply the initial contours for the first frame.

3 Template Level Sets

The next approach was to minimize the same energy (1), but to use a simplified level set function. The main idea is to use a parameterized function, such as an elliptic paraboloid, as the level set function. The segmentation contour is modified by adjusting the parameters of the level set function, for example through a line search, rather than by evolving a PDE. The hope was that this method would give results similar to the PDE-based approach, but be much faster. This idea was proposed by Richard Tsai, in an informal conversation at UCLA in June of 2003.

An elliptic paraboloid was used as the template level set because its level sets are ellipses. This template level set function is defined by:

$$\phi(x, y;\, x_0, y_0, z_0, \theta, \rho) = \bar{x}^2 + \rho\bar{y}^2 + z_0, \tag{2}$$

where

$$\bar{x} = (x - x_0)\cos\theta - (y - y_0)\sin\theta \tag{3}$$

$$\bar{y} = (x - x_0)\sin\theta + (y - y_0)\sin\theta. \tag{4}$$

A line search in the space of the parameters x_0, y_0, z_0, θ, and ρ was used to find the minimum of the energy. Although this method is attractive, in practice it is not faster than the PDE-based approach and has the disadvantage of getting trapped in local minima. These two

undesirable aspects may have been due to the line search; nevertheless, the method was abandoned. The method may, however, work well when used in conjunction with other minimization techniques, such as, genetic algorithms.

4 Frame Difference

Next I considered frame differences in a level set framework. Comparing frames, in some way, is the only way to detect motion. Frame differencing does require that either the camera is fixed or that the frames have been registered, but unlike local methods, such as optical flow estimation, is effective at low frame rates.

In this study, I considered difference images of the form

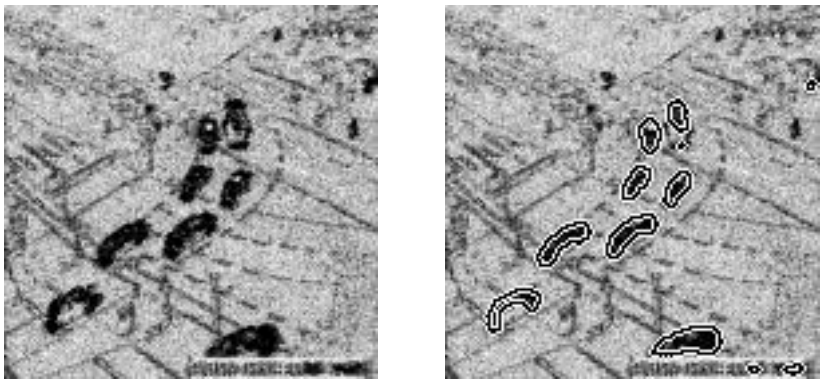
$$D_{n,m}(x) = g(\,|I_n(x) - I_m(x)|\,), \tag{5}$$

where I_n and I_m are two video frames and g is a transformation of the absolute difference. Figure 3 shows two video frames and the resulting difference frame, where $g(x) = 255\, e^{-\gamma x^2}$. This form of g has the effect of adjusting the contrast and negating the result. The difference image appears black (intensity near 0) at points were $I_n(x)$ and $I_m(x)$ are different, and appears white (intensity near 255) where $I_n(x)$ and $I_m(x)$ are the same.

(a) (b) (c)

Figure 3: Two frames of a video (a), (b), which are approximately 1/5 second apart, and their difference (c).

The difference image in Figure 3(c) has considerable noise, so that a simple threshold would give poor results. Region-based level set segmentation algorithms, such as the Chan-Vese algorithm, on the other hand, give good results because they have a regularity that emphasizes larger regions and gives a robustness to noise. Even small camera motion can be tolerated by region-based methods. Figure 4 shows the difference image from Figure 3(c) and the segmentation produced by a variation of the Chan-Vese algorithm.



(a) (b)

Figure 4: The difference image from Figure 3(c) is shown in (a), and the segmentation produced by the Chan-Vese algorithm (b).

The Chan-Vese algorithm requires the user to supply an initial contour. This initial contour is used to produce an initial level set function, which is negative in the interior of the contour, zero on the contour, and positive otherwise. The Chan-Vese algorithm makes no assumption about the intensities in these regions and iteratively determines them as it finds the best segmentation of the image.

In this application, however, there is prior knowlege about the intensities of the desired regions. In particular, the intensities of the difference image should be near 0 in the moving region and large in the background region. As a result, the user does not need to define an initial contour; and $\phi = 0$ can be taken as the initial level set function. This approach may seem unorthodox, yet it is natural in the level set framework.

5 Isolating Connected Components

The contour produced by a level set segmentation algorithm is a single object, which may have more than one connected component. Each of the connected components corresponds to one of the objects. In order to identify the individual objects, the connected components of the level set contour need to be isolated.

Figure 5 describes a simple recursive algorithm that I developed at LLNL to isolate the connected components of a level set contour. The algorithm iterates through the points of the contour, which are initially unlabeled. If a point is unlabeled, the algorithm gives it the next unused label, then calls the recursive function `find_similar` to find and label all other points that belong to the same connected component. The algorithm is fast in practice because almost all of the operations are comparisons.

```
struct point
{
   x, y
   label
 }

find_similar(point p, contour C)
{
   for each q in C
      if (q.label_not_set() and distance(p,q) <= 1){
         q.label = p.label
         find_similar(q, C)
      }
}

isolate_connected_components(contour C)
{
   for each p in C
      if (p.label_not_set()){
         p.label = next_label()
         find_similar(p, C)
       }
}
```

Figure 5: Pseudo-code of the recursive routine used to isolate the connected components of a level set contour.

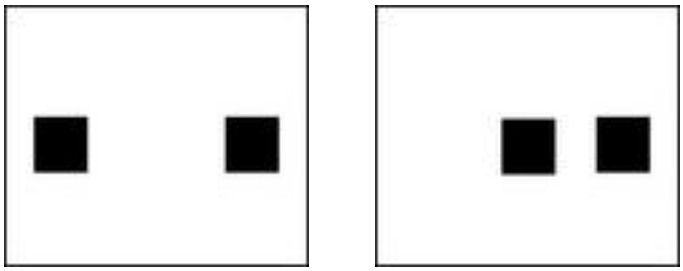
6 Three-Frame Difference

A three-frame difference is a way to handle the duplicate objects that occur with frame differencing when the differences are large. At low frame rates, the difference image often contains two instances of each moving object: one corresponding to where the object moved *to*, and the other to where the object moved *from*. This is illustrated in the cartoon sequence below. Figure 6 shows a sequence of three frames, where an object is moving from left to right.

(a) (b) (c)

Figure 6: A three frame video sequence where an object is moving left to right.

Figure 7 shows two difference images obtained from this sequence. The difference of Figures 6(a) and 6(c) is shown in 7(a) and difference of Figures 6(b) and 6(c) is shown in 7(b), where the difference is given by Equation (5), where $g(x) = 255\, e^{-\gamma x^2}$.



(a) (b)

Figure 7: The difference of Figures 6(a) and 6(c) is shown in (a) and the difference of Figures 6(b) and 6(c) is shown in (b).

By taking the intersection of these two difference images, the moving object in Figure 6(c) can be isolated, as shown in Figure 8.

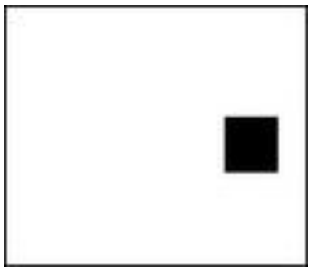


Figure 8: The intersection of the differences shown in Figures 7(a) and 7(b) recovers the moving object in Figure 6(c).

There are several ways to take the "intersection" of two images. The method that I use is to define a functional whose mininum corresponds to the desired intersection. This is described in more detail in Section 9.

7 Attempted use of the EM Algorithm

In some cases, the intersection of frame differences can contain artifacts, which are due to the aperature effect and interactions of different objects. In order improve the quality of the segmentation, I considered using the EM (Expectation Maximization) algorithm [2]. I first developed this technique while doing a term project for PSYCH 269 "Vision as Bayesian Inference", during the Spring 2003 quarter at UCLA. The main idea is to estimate the intensity distribution of an object as a mixture of Gaussians using the EM algorithm, and then to use this density to better segment the object.

In this study, I assumed that the intensities of an object (or region) were samples from a density $p(x \,|\, \mu, \sigma)$ that is a mixture (convex combination) of Gaussian densities:

$$p(x \,|\, \mu, \sigma) \;=\; \sum_{i=1}^{n} \alpha_i \, p(x \,|\, \mu_i, \sigma_i), \tag{6}$$

where $\sum_{i=1}^{n} \alpha_i \;=\; 1$, and

$$p(x \,|\, \mu_i, \sigma_i) \;=\; \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-(x-\mu_i)^2 / 2\sigma_i\right), \tag{7}$$

where μ_i and σ_i are the mean and variance of the distribution.

Given a set of sample points x_j, $j = 1, \ldots, N$ from the image, the EM algorithm is used to estimate the unknown parameters α_i, μ_i, and σ_i. Starting with initial values of these parameters the E and M steps of the EM algorithm are iteratively applied.

During the E-Step, the posterior probability

$$h_{j,i} = p(i \,|\, x_j, \mu_i, \sigma_i) = \frac{p(x_j \,|\, i, \mu_i, \sigma_i)\, p(i)}{\sum_{i=1}^{n} p(x_j \,|\, i, \mu_i, \sigma_i)\, p(i)} = \frac{\alpha_i \, p(x_j \,|\mu_i, \sigma_i)}{\sum_{i=1}^{n} \alpha_i \, p(x_j \,|\, i, \mu_i, \sigma_i)} \tag{8}$$

that data point x_j was generated from density $p(x \,|\, \mu_i, \sigma_i)$, given the current estimates of α_i, μ_i, and σ_i, is calculated.

Then during the M-Step, the estimates of α_i, μ_i, and σ_i are updated:

$$\alpha_i^{\text{new}} = \frac{\sum_{j=1}^{N} h_{j,i}}{N} \tag{9}$$

$$\mu_i^{\text{new}} = \frac{\sum_{j=1}^{N} h_{j,i}\, x_j}{\sum_{j=1}^{N} h_{j,i}} \tag{10}$$

$$\sigma_i^{\text{new}} = \frac{\sum_{j=1}^{N} h_{j,i}\, (x_j - \mu_i)^2}{\sum_{j=1}^{N} h_{j,i}}. \tag{11}$$

The use of the EM algorithm worked well for relatively large objects at higher spatial resolutions. The method, however, did not work as well for small objects and at lower resolutions. The main reason is that there were too few independent points to obtain a good estimate of the probability density $p(x \,|\, \mu, \sigma)$. This method was abandoned for the simpler method of a local background average, which is described in the next section.

8 Local Background Average

The idea behind the local background average is that the region around each object has nearly constant intensity. This is true for most surveillance video and is especially true at lower resolutions, where the image appears "blurred". Since the level set function is reinitialized by iterating the equation

$$\psi_\tau = \text{sign}(\psi)(1 - |\nabla\psi|), \tag{12}$$

the level set function has nearly a unit gradient. The region that is between ρ_a and ρ_b units away from the objects is approximately given by the set $R = \{x : \rho_a \leq \phi(x) < \rho_b\}$, which is used to calculate the average intensity of the exterior region. This method, where $\rho_a = 0$, was first used in [3]. The use of a positive ρ_a can lead to a better estimate of the exterior intensity because the objects themselves are less likely to be included. Use of the local background average is effective at removing the unwanted artifacts from the image, which are produced by the aperature effect and interactions.

9 Description of Algorithm

The final algorithm, that was developed this summer, uses a three-frame difference of images along with the local background average. These ideas are combined into a single energy (or functional) which is minimized. The difference of the current frame I_n and two preceeding frames I_{n-1} and I_{n-2} are calculated using Equation (5), where $g(x) = e^{-\gamma x^2}$ and where γ is a design parameter. The difference with the local background average is computed similarly as

$$B_n(x;\, c) = e^{-\beta(I_n(x) - c)^2}, \tag{13}$$

where c is the average intensity of the pixels in the region $R = \{x : \rho_a \leq \phi(x) < \rho_b\}$ and where β, ρ_a, and ρ_b are design parameters. For the first iteration, $\rho_a = 0$ is used, since R would otherwise be the empty set.

The image domain Ω is partitioned into interior and exterior regions. The interior region (moving objects) is modeled by the equation

$$D_{n,n-1}(x) + D_{n,n-2}(x) + B_n(x). \tag{14}$$

This expression is minimized when the differences between $I_n(x)$ and $I_{n-1}(x)$, and between $I_n(x)$ and $I_{n-2}(x)$, and between $I_n(x)$ and the local background average c are all large. The exterior region is modeled by the equation

$$(1 - D_{n,n-1}(x))\,(1 - D_{n,n-2}(x))\,(1 - B_n(x;\, c)), \tag{15}$$

which is minimized when $I_n(x)$ is close to either $I_{n-1}(x)$, $I_{n-2}(x)$, or c.

The level set function is defined as a Lipschitz continuous function ϕ, where $\phi < 0$ in the interior region and $\phi \geq 0$ in the exterior region. By using the Heaviside function H, defined by

$$H(z) = \begin{cases} 0, & \text{if } z < 0; \\ 1, & \text{if } z \geq 0, \end{cases} \tag{16}$$

and its distributional derivative $\delta(z) = H'(z)$, the complete model can be written as

$$\begin{aligned} E(\phi;\, c) = \int_{\Omega} & \lambda_{int}\,(D_{n,n-1}(x) \,+\, D_{n,n-2}(x) \,+\, B_n(x;\, c))\,(1 - H(\phi)) \\ & +\, \lambda_{ext}\,(1 - D_{n,n-1}(x))\,(1 - D_{n,n-2}(x))\,(1 - B_n(x;\, c))\,H(\phi) \\ & +\, \mu\,\delta(\phi)\,|\nabla\phi|\;dx, \end{aligned} \tag{17}$$

where $\delta(\phi)\,|\nabla\phi|$ is the length of the boundary of the interior region. More details about this approach can be found in [1].

The functional given in Equation (17) is minimized by first computing the Euler-Lagrange equations and then solving the resulting PDE. In order to compute the Euler-Lagrange equations, regularized versions of H and δ are used. In this algorithm, the regularizations proposed by Chan and Vese [1] are used. These are given by

$$H_\epsilon(z) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{z}{\epsilon}\right), \tag{18}$$

$$\delta_\epsilon(z) = \frac{\epsilon}{\pi\,(z^2 + \epsilon^2)}. \tag{19}$$

Applying these regularizations to Equation (17) and computing the Euler-Lagrange equations leads to the following PDE:

$$\begin{aligned} \frac{\partial\phi}{\partial t} = \delta_\epsilon(z)\,(\lambda_{int}\,(D_{n,n-1}(x) + D_{n,n-2}(x) + B_n(x;\, c)) - \lambda_{ext}\,(1 - D_{n,n-1}(x)) \\ \cdot\,(1 - D_{n,n-2}(x))\,(1 - B_n(x;\, c)) + \mu\,\mathrm{div}(\frac{\nabla\phi}{|\nabla\phi|})) \quad \text{in } \Omega, \end{aligned} \tag{20}$$

$$\frac{\partial\phi}{\partial n} = 0 \quad \text{on } \partial\Omega, \tag{21}$$

$$\phi_0 = 0 \quad \text{in } \Omega. \tag{22}$$

10 Sample Results

This algorithm is illustrated with two video sequences from a well-known image sequence database [6]. In each of the examples, the sequences were sampled at one frame-per-second before being used in the algorithm. Three frames of each sequences are shown, along with the segmentation of the third frame at varying spatial resolutions. In both cases, the algorithm is able to detect the moving objects at considerably low resolutions. The low resolution images were produced by averaging pixels from the higher resolution image.

(a) (b) (c)

Figure 9: A three frame video sequence [6] sampled at 1 frame-per-second.

Figure 10: Segmentation of Figure 9(c), where the spatial resolution has been reduced 16x (a), 64x (b), 144x (c), and 256x (d).

(a) (b) (c)

Figure 11: A three frame video sequence [6] sampled at 1 frame-per-second.

Figure 12: Segmentation of Figure 11(c), where the spatial resolution has been reduced 16x (a), 64x (b), 144x (c), and 256x (d).

In these examples, the algorithm produced very good results at a 64x reduction in spatial data. Since the sequences were processed at 1 fps instead of the original 24 fps, the total reduction in video data is $24 \cdot 64\text{x} = 1536\text{x}$. This is a reduction in data from gigabytes to megabytes and from terabytes to gigabytes. This reduction in data not only relieves storage requirements, but also speeds computation since less data needs to be moved in and out of memory and processed. As a final note, the algorithm is not restricted to 1 fps and should work at even lower frames rates, the limitation is not in the segmentation per se, but in the frame-to-frame correlation of the data.

11 Closing Remarks

The algorithm described in Section 9 is considerably effective and robust. It does, however, have its limitations. Similar to methods that use background subtraction, the algorithm does not perform well when the moving objects have the same intensity as the background. The algorithm also does not distinguish between the moving objects and their shadows, which may or may not be a problem in a given application. The algorithm assumes that the objects have an apparent motion from one frame to the next and does not detect objects with a low line-of-sight rate relative to the camera. Since the algorithm is assumed to be used for surveillance, where the camera is at a distance above the objects, this is not expected to be a problem.

There are several directions to take in extending this algorithm. The most significant would be to develop an algorithm that provides a frame-to-frame correlation of the objects and to develop the data mining algorithms for which this algorithm was intended.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by the University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

References

[1] T. Chan and L. Vese. An active contour model without edges. *Int. Conf. Scale-Space Theories in Computer Vision*, 16(2):266-277, 1999.

[2] Dempster, A.P., Laird, N.M., and Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm (with discussion), *Journal of Royal Statistical Society, Ser. B* 39:1-38, 1977.

[3] M. Moelich and T. Chan. Tracking Objects with the Chan-Vese Algorithm. *UCLA CAM Reports*, 03-14, 2003. `http://www.math.ucla.edu/applied/cam/index.html`.

[4] J. Morel and S. Solimini. *Variational methods in image segmentation*, Birkhäuser, 1995.

[5] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems, *Comm. Pure Appl. Math.* 42, 1989.

[6] H. Nagel and Mitarbeiter. *KOGS/IA KS Universitat Karlsruhe* (Image sequence database). `http://i2lwww.ira.uka.de/image_sequences`.

[7] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, Springer, 2003.

[8] J. Sethian. *Level set methods and fast marching methods*, Cambridge University Press, 1999.