Spatio-Temporal Detection and Isolation: Results on PETS 2005 Datasets

Richard Souvenir
Computer Science and Engineering
Washington University
St Louis, MO 63130

John Wright
Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Robert Pless
Computer Science and Engineering
Washington University
St Louis, MO 63130

Abstract

Recently developed statistical methods for background subtraction have made increasingly complicated environments amenable to automated analysis. Here we illustrate results for spatio-temporal background modeling, anomaly detection, shape description, and object localization on relevant parts of the PETS2005 data set. The results are analyzed both to distinguish between difficulties caused by different challenges within the data set, especially dropped frames, recovery time from camera motions, and what can be extracted with very weak assumptions about object appearance.

1 Introduction

Automated surveillance algorithms are most useful if the conditions under which they are effective are well understood. Comparing different algorithms for real-world scene analysis is challenging because different applications assume different constraints or prior assumptions on the environment. The performance of an algorithm then depends on the implementation, the choice of representations, and whether or not the environment being imaged fits the assumptions implicit in the algorithm. Since the current set of environments amenable to automated analysis is small, there is value in new algorithms demonstrating capabilities on new data sets in new environments. The PETS data sets serve to highlight specific environmental conditions and allow comparative analysis.

The 2005 PETS data sets [1] contain surveillance video of coastal regions. The video is comprised of jpeg compressed frames whose background includes shore, waves, and sky, and whose foreground may include small and large objects on water. The data is challenging in several respects: (a) the view is not stationary, but pans-tilts-zooms to new locations requiring the analysis of background motion to adapt to new situations, (b) the images are significantly compressed, noisy, and the video includes dropped frames, and (c) in some scenes the objects to be detected are quite small.

A collection of statistical techniques that fall under the rubric of spatio-temporal video analysis have recently been proposed for the analysis of video data. These techniques capture spatio-temporal statistics of the video and identify pixels or regions that do not fit this statistical model. These techniques have been applied globally, using PCA decomposition of the image sequence to create a set of basis functions to reconstruct the background [13], but it is not clear that relevant surveillance scenes with natural background motions are well modeled with a small set of linear basis functions.

The opposite end of the spatio-temporal analysis spectrum includes models that are local at each pixel, representing either the intensity and variance at each pixel [9], or the local optic flow and its variance [7]. These models suffer, respectively, from the inability to identify anomalous objects at locations with large intensity changes caused by consistent motions, and the classical challenges with computing optic flow in real environments.

Our approach, VAnDaL (Video Anomaly Detection and Localization) is to model the joint distribution of spatio-temporal derivative values at each pixel; these measurements are well defined in each frame, the distribution captures the statistics of consistent motions at a pixel, and empirical evaluation has in the past indicated much better performance in many different environments over pixel based intensity models [8]. Furthermore, recent work shows that these local models at each pixel can be cleanly clustered to capture consistent motion patterns across the entire scene in, for example, traffic scenes with different global motion pattern [12].

This paper explicitly studies two questions. The first is: How effective are spatio-temporal derivative models in capturing the statistics of the background water motion in the PETS2005 database? In particular, what is the effect of dropped frames and how does the time over which the model is accumulated affect the model?

Robust image derivative measurements are estimated using filters of large support, so when objects are detected, their position may not be accurately identified. This leads to the second question: Given an approximate model of an unknown object in a scene with a complicated background, are there natural statistical techniques to improve the accuracy of the object position estimate and to define a more specific figure-ground segmentation? Our main results are the following:

- First, dropped frames in a video have a significant impact on the performance of local spatio-temporal background models. Automatically detecting the dropped frames and discarding corrupted temporal derivatives results in good models of the background, at the cost of missing anomalous objects appearing in frames before and after the dropped frames. If the temporal derivatives are rescaled to account for the dropped frames, then the specificity of the background model recovers partially, and does not miss the analysis of any of the frames present.

- Second, local spatio-temporal background models can effectively accommodate applications that require different time scales of adaptation. Furthermore, they can be fielded effectively on cameras which pan, tilt, and zoom, as they can rapidly adapt to the new background when the camera stops at a new viewing direction.

- Third, simple EM style algorithms can refine the spatio-temporal anomaly detection results, to provide a very accurate object position, shape, and orientation, and appearance without strong priors on object appearance.

The next section provides a brief introduction to the spatio-temporal background modeling methods used, followed by results illustrating the quality of the background model for different scenes using different time-decay constants, and in conditions with and without dropped frames. The following section considers the problem of more accurately localizing the anomalous object.

2 Spatio-Temporal Background Modeling

This section gives the mathematical framework for one method of spatio-temporal background modeling. The method works by keeping a model, at each pixel, of the joint distribution of the x, y, t derivatives of intensity. The method described has been implemented in a real time system that runs on an 800 MHz laptop for image resolutions up to 640 by 480, using openCV and the Intel Image Processing Library.

2.1 Spatio-Temporal Structure Tensor Field

Let $\nabla I(\vec{p}, t) = (I_x(\vec{p}, t), I_y(\vec{p}, t), I_t(\vec{p}, t))^T$ be the spatio-temporal derivatives of the image intensity $I(\vec{p}, t)$ at pixel $\vec{p}$ and time t. At each pixel, the structure tensor, Σ, is defined as

$$\Sigma(\vec{p}) = \frac{1}{f} \sum_{t=1}^{f} \nabla I(\vec{p}, t) \nabla I(\vec{p}, t)^T$$

where f is the number of frames in the sequence and $\vec{p}$ is omitted after this for clarity's sake.

To focus on scene motion, the measurements are filtered, only considering measurements that come from motion in the scene, that is, measurements for which $|I_t| > k$, so that the model only considers the spatio-temporal derivatives with significant temporal change (in the experimental section, we choose $k = 10$, requiring that a pixel intensity change by 10 grayscale values (out of 255) for the derivative measurements to be testbed by or incorporated into the model). Furthermore, we assume the mean of ∇I to be zero (which does *not* imply the motion is 0). Under this assumption, Σ defines a Gaussian distribution $\mathcal{N}(0, \Sigma)$. Anomalous measurements can then be detected by comparing the Mahalanobis distance, $\nabla I^T \Sigma^{-1} \nabla I$ to a preselected threshold [8]. For anomaly detection, then, this is a two part model, that classifies a pixel as belonging to the background if either (a) $I_t < 10$, or (b) $\nabla I^T \Sigma^{-1} \nabla I <$ threshold. This threshold needs to be set to accommodate application specific trade-offs between allowable false positive and false negative rates.

The structure tensor is related to optic flow. If a pixel always views the same optic flow u, v, then all derivative measurements exactly fit the optic flow constraint equation: $I_x u + I_y v + I_t = 0$ [5], and the structure tensor has rank $\leq$ 2. Its third eigenvector is a homogeneous representation of the total least squares estimate of the optic flow [3, 6, 11].

Under the assumption of stationarity, Σ can be estimated online as the sample mean of $\nabla I \nabla I^T$. For non-stationary distributions, the model can be allowed to drift by instead assigning a constant weight, $\alpha \in [0, 1]$, to each new measurement:

$$\Sigma_t = (1 - \alpha)\Sigma_{t-1} + \alpha \nabla I \nabla I^T.$$

This update method causes the influence of a given measurement on Σ to decay exponentially, with decay constant $\frac{-1}{ln(1-\alpha)}$. Section 2.2 investigates the effect of specific choices the forgetting factor, α. The results in the following section mostly focus on the statistics of the spatio-temporal structure tensor. This has been used to detect anomalies in the video by comparing the Mahalanobis distance, $\nabla I^T \Sigma^{-1} \nabla I$ to a preselected threshold [8].

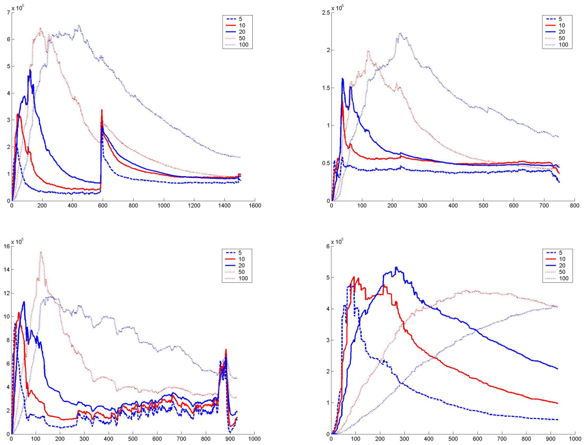


Figure 1. Plots of the determinant of Σ accumulated over time with different decay factors (starting from the top left, corresponding to zod5a, zog6, zod7a, zod9). The x-axis is the frame number, and the y-axis is the mean, over the whole image, of $\det \Sigma$.

Defining a threshold allows the covariance matrix to be used as a classifier to determine if a current x,y,t derivative triple is background or foreground. For a given threshold, τ, all measurements lying inside the equi-probability ellipsoid, $\varepsilon = \{\nabla I : \nabla I^T \Sigma^{-1} \nabla I = \tau\}$, will be judged to come from the background distribution. The volume of this ellipsoid is $\frac{4}{3}\pi\tau^{\frac{3}{2}}|\Sigma|^{\frac{1}{2}}$. Suppose the foreground distribution is taken to be uniform over a region $V \subset \mathcal{R}^3$ containing ε. Then the probability of misclassifying a measurement generated by an anomalous object is $\frac{4\pi\tau^{\frac{3}{2}}|\Sigma|^{\frac{1}{2}}}{3 \times volume(V)}$, giving $P(\text{false negative}) \propto \sqrt{\det \Sigma}$. Because the probability of misclassifying a background pixel is related to $\det \Sigma$, this value serves as a summary statistic, which can be used to evaluate the likelihood that foreground pixels will be inaccurately classified as background model, without considering a specific threshold.

2.2 Results

This section considers the variation in the background model specificity as a function of the time constants used in the creation of the model, the data in the scene, and dropped frames. We give a brief description of methods to explore each source of variation and discuss the findings one by one.

Time decay constants An important free variable in defining the behavior of the spatio-temporal structure tensor is the decay factor. This factor defines the effective time window used to generate the current model. For a time constant, c, the influence of a given measurement from t-frames earlier falls as $\exp(-t/c)$. The data at the output is 63% determined by the past c frames. For different time constants, we are interested in the specificity of the background models. We consider the determinant of the covariance matrix as an indicator of this specificity.

Figure 1 shows the effect of different choices of time constants on the convergence of the model. The spike near frame 600 is caused by a man walking across the screen in a way that violates the background model. In practice, measurements judged to come from foreground objects can be excluded from the model.

Discussion. Several interesting features come to light; first the rate of convergence to a background model depends on both the decay rate and the scene. Second, even at steady state, the determinant of Σ is smaller for a time constant of 5 but roughly equal for time constants of 10 and 20, indicating the background model distribution is non-stationary for very short image sequences.

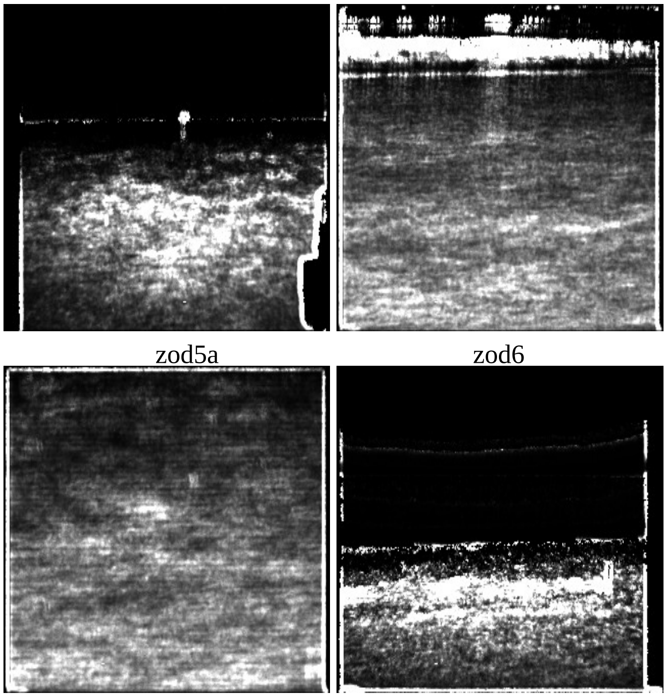
Data dependence The effectiveness of the background model also depends on the characteristics of the scene. Since the spatio-temporal background models we consider are local, the effectiveness depends upon the local scene appearance. Figure 2 attempts to characterize the specicificity of the background models for difference image regions. These maps are made as the $\det \Sigma$ value of the pixels over the whole video, using a time constant of 100. The evolution of the the model over two specific pixels is shown in Figure 3.

Discussion. These image maps indicate several important properties of the spatio-temporal background model.

The zod5 image has a larger $\det \Sigma$ in the image center on the water than the surrounding water areas. This is caused by the higher contrast in that part of the video; insofar as this image contrast variation is due to the camera, the contrast would also be higher in this area for anomalous objects, so the actual sensitivity to objects (as opposed to sensitivity to image derivatives not fitting the model) may be approximately constant. The bright spot at the waterline is due to variations caused by significant compression artifacts that change the appearance of the bright white signpost. The bright edge shown on the bottom right corner is due to small camera motions and a very high contrast edge in the scene.

The zod6 video illustrates two features. First, on the water, the model is less specific in the areas near the camera where the largest and most varied wave motion appears. Additionally, the $\det \Sigma$ score is large in the sky region — this region sees large and varied motion at the beginning of the sequence as the camera pans past a large ship, after which there is no variation in the area so the background model is not updated.

The zod7a video consists of many different camera views and the camera pans to keep a boat approximately in the



zod5a zod6

zod7a zod9

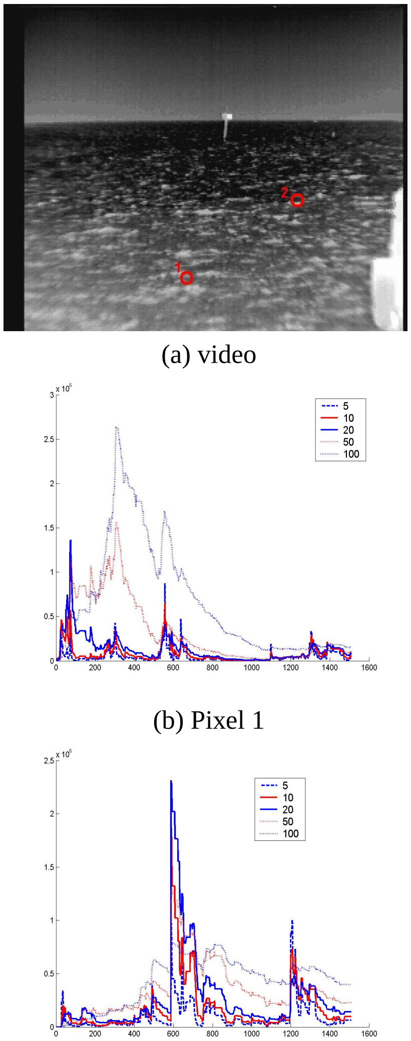
Figure 2. Plots of the determinant of Σ at each pixel. Bright white corresponds to $\det\Sigma \geq 2.55 \times 10^5$.

field of view. Effective analysis of this video probably requires algorithms to explicitly detect camera motion to reset the background background model at each new static viewing direction.

The zod9 video is much like the zod5a video, with the same pattern of of higher contrast in the middle of the image. The large $\det\Sigma$ values along the horizon line are caused by compression artifacts at this high contrast edge.

Dropped Frames Finally, we consider the effects of temporal discontinuities; dropped frames in the video sequence. These have the potential to significantly affect spatio-temporal methods, because the difference between consecutive frames is no longer an estimate of the temporal derivatives. Figure 4 shows the mean $\det\Sigma$ in three cases, (a) when the algorithm ignores the fact that some frames are missing (just analyzes the video as is), (b) when the algorithm ignores all derivative estimates corrupted by missing data, and (c) when the algorithm rescales the estimated temporal derivative to account for the missing frame.

Discussion. The effect of temporal discontinuities on convergence of the mean of $\det\Sigma$. The red (middle) line shows results with discontinuities removed. Note the qualitative difference in convergence behavior, as well as the doubling

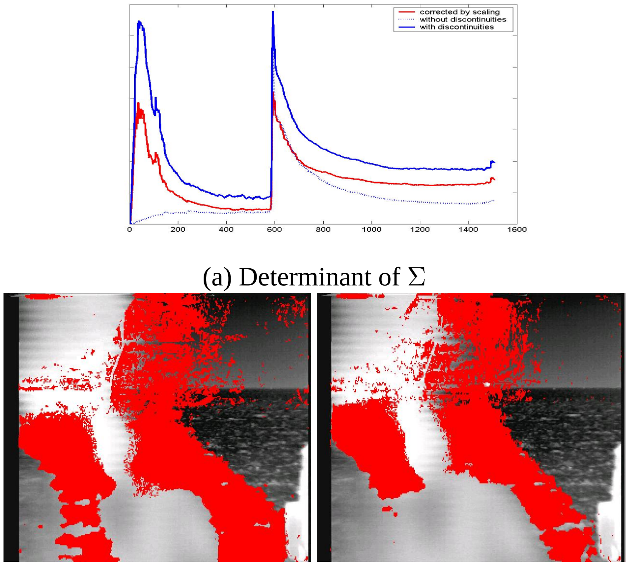


(a) video

(b) Pixel 1

(c) Pixel 2

Figure 3. Plots of the determinant of Σ over time for two particular pixels in the video zod5a. Axes are labeled as in Figure 1. The larger magnitude derivatives at pixel 1 lead to a more stable model.



(a) Determinant of Σ

(b) Without Discontinuities (c) With Discontinuities

Figure 4. The effect of temporal discontinuities on convergence of the mean of $\det \Sigma$. The red line (the middle of the three lines) shows results with discontinuities removed. (b) and (c) show the effect on anomaly detection, pixels where the Mahalanobis distance exceeds 15 are marked. More of the object is marked when discontinuities are removed from the model.

of $P(\text{false negative})$ in the steady state. (b) and (c) show the effect on anomaly detection. In each, pixels where the Mahalanobis distance exceeds 15 are marked. The lower left corner of (c) shows the failure to mark the shoulder of a passing person because of the weaker model.

2.3 Lessons Learned

Real world applications demand robustness to video. Although spatio-temporal methods are affected by noisy data, they may still be effective despite temporal discontinuities and dropped frames (*zod5a* and *zod9*), gross quantization artifacts (*zod9*), and interlacing artifacts in fast moving sequences (*zod6a* and *zod7*).

The next section considers what might be a subsequent step in a complete surveillance system, the analysis of a large set of image regions identified as being anomalous.

3 Localization and appearance modeling

The results of background subtraction often comprise an incomplete and inaccurate segmentation of the figure or object from the background. This is especially true with spatio-temporal background models that use filters with large support to make robust image measurements. More accurate isolation of the object position and refinement of object appearance would facilitate higher level object recognition. This section considers the following problem:

> **Problem**: Given a sequence of images, and a sequence of approximate locations of an object within each image, find a generalized object model and the exact object location and orientation in each frame.

We consider this problem in the context of the *zod6* video data set. A representative collection of images of the zodiac and the immediately surrounding area is shown in Figure 5.

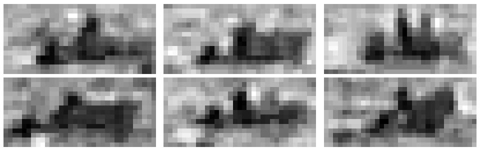


Figure 5. Six frames approximately centered of the zodiac boat.

Given that there is no explicit model for the object being tracked and each frame only contains low-quality, low-resolution representations of the object, we make the following assumptions to make this problem tractable for the PETS zodiac boat data set zod6. First, we assume that the background intensity (the intensity of the water) is drawn from a stationary distribution. Second, we assume that the shape of the foreground object is a rigid 2D shape that may translate and rotate. Third we assume that the intensity of the foreground object (the boat) is drawn from a different distribution than the background (the water). We feel that these, or similar, assumptions may apply to a broad category of detection and isolation problems in surveillance.

3.1 Related Work

This problem falls between the classifications of segmentation, tracking, and modeling of texture and shape. There is a collection of work that studies related problems. Representative papers include the blobworld approach using EM to segment regions of individual images based on texture and color, and discovering potential objects or image regions useful in image retrieval [2]. For tracking, an online formulation uses EM to segment cars in aerial video, with a primary goal of improving tracking in multi-object scenes [10]. Our problem differs from these works in that (a) unlike the first paper, we segment the object based on many frames instead of just one, and (b) unlike the second paper, our goal is to optimize the appearance model of an object that is visible over hundreds of frames.

3.2 Refining Localization

In this section, we formalize the problem and describe our procedure to iteratively refine the appearance and location of tracked objects in a scene.

Given a set of images $\mathcal{I} = \{I_1, I_2, \ldots, I_N\}$ and approximate locations of an object in the corresponding image $\hat{\mathcal{X}} = \{\hat{x_1}, \hat{x_2}, \ldots, \hat{x_N}\}$, find the exact object location, $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ and orientation $\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$ in each frame.

In our problem domain we consider the object to be a 2D rigid object, parameterized, in frame i, by position x_i and rotation θ_i, although our approach allows more general transformation models. Note that we do not have a prior model of the appearance or shape of the boat. We assume the intensity of each pixel on the object is drawn from a Gaussian distribution and the intensity of pixels in the background are also drawn from a different Gaussian distribution.

We can now describe this as a learning problem, where we observe the pixel intensities and the foreground/background segmentation in each frame is the hidden data. Therefore, using an iterative-updating method such as the Expectation-Maximization (EM) algorithm [4] is a natural choice. The framework of the algorithm is the following is shown in Algorithm 1, which we elaborate on in this section.

The goal of this algorithm is to estimate the segmentation and the texture of the foreground. This information is captured in the image template, M. Figure 6 shows a sample template for the boat in the zod6 dataset. The choice of initialization of the template affects the tendency of this EM algorithm to converge to local maxima. The template can be randomly initialized or domain-specific knowledge can be used to bootstrap the process. In the case of the zodiac dataset, we initialize the template with a rectangle drawn around the region detected by the spatio-temporal background model. The size of this rectangle was chosen by hand in the experiment, but we assume no knowledge of the shape of the boat within the rectangle. Figure 8 shows the refinement of the template over the course of the EM algorithm, and the first image shows the template used for initialization in Algorithm 1.

Lines 5-8 describe the estimation steps of our algorithm. For each region of interest in a frame, we estimate the position and appearance of the object using the template as a filter. Given our assumption that the shape of the foreground is constant up to a rigid transformation in the image, we convolve the template with the frame through a range a rotations to find the location of highest response.

Algorithm 1 EM_Localization($\mathcal{I}, \hat{\mathcal{X}}$)

```
1:  Initialization:
2:  For each image $I_j \in \mathcal{I}$ create a set of equal-sized regions of interest $\mathcal{I}^{ROI} = \{I_1^{ROI}, I_2^{ROI}, \ldots, I_N^{ROI}\}$ around the approximate locations $\hat{x}_j \in \hat{\mathcal{X}}$
3:  Create an image mask $M$ which contains foreground/background segmentation and image texture values for foreground pixels.
4:  repeat
5:      Estimation Step:
6:      for all $I_j^{ROI} \in \mathcal{I}^{ROI}$ do
7:          Using image mask $M$, estimate the location, $x_j$, and appearance, $\theta_j$, of the object in $I_j^{ROI}$
8:      end for
9:      Maximization Step:
10:     Warp each image $I_j^{ROI} \in \mathcal{I}^{ROI}$ to align the putative foreground pixels
11:     Recalculate the image mask, $M$, to maximize the likelihood of $\mathcal{X}$ and $\Theta$, given $\mathcal{I}^{ROI}$.
12: until image mask, $M$, converges
```

In order to maximize the likelihood that the positions, $\mathcal{X}$, and appearances, Θ, estimated in line 7 represent the foreground pixels given the images, $\mathcal{I}^{ROI}$, we construct a *transformed stack (T-stack)*. The T-stack has size $h * w * N$ where h and w are the height and width, respectively, of the region of interest and N is the number of images. For each region of interest, $I_j^{ROI} \in \mathcal{I}^{ROI}$, we create a corresponding image in the T-stack through translation by $(\hat{x}_j - x_j)$ and rotation by $-\theta_j$. The T-stack registers every image such that the pixels falling within the template — the putative foreground pixels — are aligned along the 3^{rd} dimension. Figure 7 shows two example regions of interest, a binary image mask defining the valid template region, and the result of warping these two images so that they are aligned in the T-stack.

In order to update the template, we now consider each pixel in the template model and the vector of all T-stack values at that pixel (some pixels may not currently not be part of the template if they were judged to not be foreground in a previous iteration; we consider these pixels in this step as well). For each vector we calculate the mean and maximum a posteriori (MAP) estimate. To segment each vector as foreground versus background we return to our assumption that the background, while dynamic, is regular and drawn from a stationary distribution. Moreover, we assume the foreground pixels are drawn from a distinct distribution. We employ well-known Gaussian mixture model clustering techniques to assign each vector to one of the two distributions and create the binary segmentation mask. To build the most likely texture model for the image template, we use the MAP estimate of intensity values from the vectors in the T-stack in the corresponding foreground pixels of the image template. Figure 6 shows an example of textured image template.

This iterative procedure refines the image template until convergence. That is, the procedure terminates when the classification of T-stack vectors does not change for consecutive iterations. We then return the final estimate of the image mask, M, which represents our belief of the appearance of the tracked object, and the positions, X, and appearances, Θ, of the object in each frame.

3.3 Results

Figure 8 shows the convergence of the model over 8 iterations of the algorithm, when using the longest subsequence (about 600 frames) where the camera was static. The image template was initialized using a small rectangle around the boat in the 300th frame. The convergence is quite rapid.

One question to ask is how many frames are necessary for this learning approach to solve for both the shape and appearance of the boat. To address this question, we ran the EM-algorithm on 5, 25, 75, 200, and the entire 640 frame subsequence. The computed template on which the algorithm converged is shown in Figure 9. The algorithm uses very weak priors on object appearance and therefore requires many frames to converge — even using a 200 frame subsequence there are pixels far from the boat that appear in the final template.

Using the template derived from the 600 images, the computed rotation and orientation fits, to visual inspection, exactly on the boat in every frame (justifying the assumption that the object is well modeled by a rigid motion within the image). The position and is captured to an accuracy at least as good as the provided ground truth data, and there is no orientation provided in the ground truth. A video sequence shows the results of this tracking and is available on request.

4 Conclusions

This report provides a performance characterization for a collection of spatio-temporal background modeling tools for the PETS2005 coastal data set. We consider both spatio-temporal model of dynamic background appearance, and the subsequent, more specific isolation and localization of independent objects. Our conclusions may be summarized as follows:

- Local spatio-temporal models are effective for current surveillance video, in spite of significant compression artifacts, noise, and frequent pan-tilt-zoom operation of the camera.

- The affect of the current environment on the specificity of the model can be characterized and displayed do determine if the algorithm is likely to be effective (see Figure 2).

- Local Spatio-temporal background models can accommodate dropped frames, although performance degrades significantly if the algorithm does not explicitly check for this condition and correct the temporal derivatives.

- Finally simple EM type algorithms using very weak assumptions about object appearance can refine the spatio-temporal anomaly detection results, to provide a very accurate object position, shape, and orientation, and appearance.

References

[1] T. Boult. Pets2005 costal surveillance datasets. http://pets2005.visualsurveillance.org/.

[2] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Color- and texture-based image segmentation using em and its application to image querying and classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[3] K. Daniilidis and H.-H. Nagel. Analytical results on error sensitivity of motion estimation from two views. *Image and Vision Computing*, 8:297–303, 1990.

[4] A. Dempster, N. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–38, 1977.

[5] B. K. P. Horn. *Robot Vision*. McGraw Hill, New York, 1986.

[6] S. V. Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, 1991.

[7] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *CVPR*, 2004.

[8] R. Pless, J. Larson, S. Siebers, and B. Westover. Evaluation of local models of dynamic backgrounds. In *CVPR*, 2003.

[9] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

[10] H. Tao, H. S. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(1):75–89, 2002.

[11] J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. *International Journal of Computer Vision*, 14:67–81, 1995.

[12] J. Wright and R. Pless. Analysis of persistent motion patterns using the 3d structure tensor. In *IEEE Workshop on Motion and Video Computing*, 2005.

[13] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *ECCV*, pages 44–50, 2003.

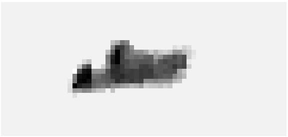


Figure 6. A textured image template for the zodiac boat image dataset.

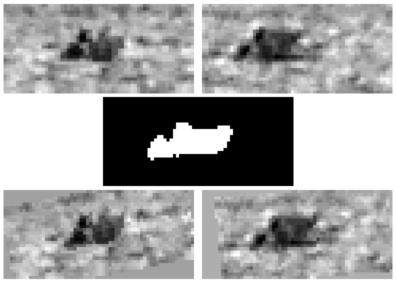


Figure 7. Top row shows example regions of interest. The middle row shows a binary image mask where the white pixels represent the foreground. Bottom row shows the result of warping the top row so that the pixels of the image template fit the texture model of the boat.

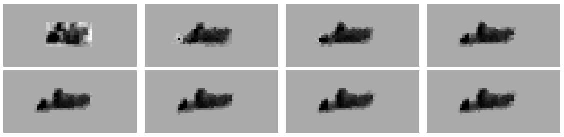


Figure 8. These images show the refinement of the image template as the EM algorithm converges.

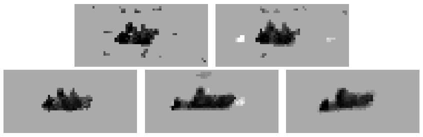


Figure 9. The final template on which the EM algorithm converges, for video clips of 5, 25, 75, 200, and 640 frames. Note that even 200 frames includes pixels that are not on the real object, arguing the developing these models over very long image sequences (when available) is important.