Into the woods: visual surveillance of non-cooperative and camouflaged targets in complex outdoor settings

Terrance E. Boult* Ross J. Micheals Xiang Gao Michael Eckmann
Vision And Software Technology (VAST) Laboratory, Lehigh University

Keywords: TV surveillance systems Image motion analysis Image sequence analysis Adaptive signal processing Real time systems, Hysteresis Omnidirectional video Connected Components

Abstract

Autonomous video surveillance and monitoring of human subjects in video has a rich history. Many deployed systems are able to reliably track human motion in indoor and controlled outdoor environments, e.g. parking lots and university campuses. A challenging domain of vital military importance is the surveillance of non-cooperative and camouflaged targets within cluttered outdoor settings. These situations require both sensitivity and a very wide field of view and therefore are a natural application of omni-directional video.

Fundamentally, target finding is a change detection problem. Detection of camouflaged and adversarial targets implies the need for extreme sensitivity. Unfortunately, blind change detection in woods and fields may lead to a high fraction of false alarms, since natural scene motion and lighting changes produce highly dynamic scenes. Naturally, this desire for high sensitivity leads to a direct tradeoff between miss detections and false alarms.

This paper discusses the current state-of-the-art in video-based target detection, including an analysis of background adaptation techniques. The primary focus of the paper is the Lehigh Omnidirectional Tracking System (LOTS) and its components. This includes adaptive multi-background modeling, quasi-connected components (a novel approach to spatio-temporal grouping), background subtraction analyses, and an overall system evaluation.

1 Introduction

There have been many visual surveillance and tracking systems developed with a variety of software and hardware architectures. We first present a brief introduction to visual surveillance systems and their overall system architecture. The system architecture overview will provide a framework in which to discuss prior work and the domain constraints. Thus citations to and discussions of related work are omitted in this secction but but can be found in the section 1.3 and throughout the remainder of the paper. This section also defines some of the terms and basic concepts used in the remainder of the paper. Following the fundamentals, we discuss difficulties in section 1.2 and then provide an overview of the remainder of the paper in section 1.3. Those already familiar with the field may wish to skip to the paper overview.

*This work supported in part by DARPA VSAM program and ONR MURI program. Contact author tboult@eecs.lehigh.edu

1.1 Visual Surveillance Fundamentals

The visual surveillance problem pertains to the the use of imaging sensors to monitor the activity of targets in a scene. For example, monitoring human activity in office environments via CCTV cameras, performing surveillance on vehicles at night with long-wave infrared sensors, or tracking soldiers in the woods using omnidirectional video are all visual surveillance problems.

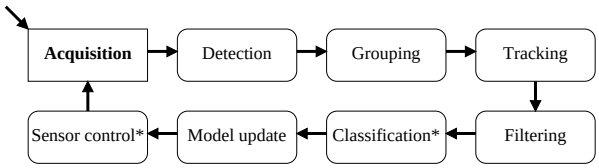

Figure 1: Logical decomposition and control flow of a visual surveillance and tracking system.

A simple view of the visual surveillance problem has two major subproblems; targets must first be *detected* and then *tracked*. From an architecture point of view, many visual surveillance systems further decompose the problem resulting in upto eight major stages: **acquisition, detection, grouping, tracking, filtering, classification, updating models,** and **sensor control**. A functional decomposition of a system can be seen in Figure 1. In some systems, a few of these stages may be performed simultaneously or may be omitted (those marked with *'s), but for illustrative purposes, we will consider each stage separately. We will briefly review these fundamental steps of visual surveillance in the context of a *reference subtraction* paradigm. Note that this is a very simplified view of the system and in the remainder of the paper, difficulties and limitations inherent in this simple view will be discussed and overcome.

Undetected targets cannot be tracked. Therefore, the first steps in a visual surveillance system are acquisition of an image and *detection* (referred to in this paper and other literature, as the change detection phase). The simplest detector based on the reference subtraction model uses a single reference image B, which is also known as a background image. The reference image is used to capture information that is "unimportant" (static) in a scene. The chief advantage of a reference modeling approach is that there is no need to explicitly model either the geometry or photometry of a scene. This not only significantly simplifies the system's operation, but allows it to operate in a wide variety of environments without the need for complex calibration routines.

Reference models operate on a very simple principle. Suppose we point a camera at a scene we wish to monitor and assume no targets are currently populating the scene. If we save this captured image as our reference image B, then any change in this scene can be detected by simply performing a pixel-by-pixel comparison between our reference image B and any new incoming image. Suppose we have a new image of our scene I that contains a target of interest. Then, we construct a *difference* image, $\Delta = |B - I|$, where each pixel of Δ is the absolute value of the difference between corresponding pixels of B and I. Therefore, each pixel of Δ that is greater than some noise-based threshold τ represents a pixel that has changed due to the presence of a target. In this paper, and in much of the visual surveillance literature, such pixels are called *target pixels*. These target pixels are the output of the detection phase. Naturally, there is a sensitivity tradeoff corresponding to different choices of τ. It must be high enough to ignore noise, but low enough to detect targets. There are two primary approaches taken to reference modeling. One adapts the reference image over time by blending information and statistics extracted from the many images. The other method uses the two most recent frames for building the difference image Δ. Both techniques have their own advantages and disadvantages.

Let us assume our simple tracker has completed the detection stage, and has moved into the *grouping* stage. It is the goal of this stage to assign a label to each potential target pixel. Ideally, all pixels that belong to the same physical target will share the same label. The most basic form of grouping is simple

connected components — the assignment of the same label to adjacent[1] pixels. Because this stage is early in the system, these uniformly labeled and well-connected pixel regions are not necessarily deemed true targets. They are often referred to with the less descriptive term *blobs*. Some systems "clean" these blobs by performing binary morphology. This helps eliminate very small regions, which are assumed to be noise, and connects together regions with small separations. Unfortunately it also deletes small targets.

In the *tracking* phase, each blob is associated with zero or more blobs computed from previous frames. Each set of spatio-temporal groupings, or *tracks*, describes a target's behavior and properties over time. The most rudimentary update consists of associating blobs that either overlap spatially with blobs from previous frames, or have centroids that are within some proximity threshold of each other. Ideally, each track corresponds to a single moving target as it moves through a scene, even if the target becomes temporarily occluded (partially or fully).

In the *filtering* stage, additional testing is performed on the target regions to insure that they are targets. Processing can vary from the simplistic deletion of targets without a minimal number of pixels on target, to the more complex methods for detecting blobs that result from reflections, shadows, or other illumination changes. For example, it is common for human trackers to assume that their targets will be upright. Such trackers may eliminate blobs that do not fit this description.

Many high-level surveillance systems augment their trackers with target recognition routines. This optional *classification* stage usually follows or is concurrent with filtering. It is optional in the sense that many visual trackers, including LOTS, do not have this component. Naturally, the granularity and type of classification vary across different systems. For example, some systems may try to distinguish between vehicle and non-vehicle targets. Others may attempt to identify a particular target's identity, or simply try to decide if the target has previously been tracked.

In the *model update* stage, the system updates its internal models to incorporate information gained in the new frame. This might include adjusting various thresholds, adjusting the background model, or updating other internal system variables. Since in many systems there can be several parameters per pixel, there are often a very large number of parameters to consider. Creating a system that both properly and rapidly self-adapts its internal model is perhaps the largest challenge faced by the visual surveillance community.

Finally, as a result of a model update, a system may decide to adapt the incoming video stream in the *sensor control* stage. Simple updates may be, for instance, simply changing the brightness or contrast of the video. An active vision system might cue a pan-tilt-zoom camera so that it may follow a tracked target.

1.2 Difficulties

We now briefly touch upon some of the difficulties faced by the above tracker and even the most state-of-the-art tracking systems. The fundamental difficulty of change detection, naturally, lies in the fact that scenes, even in controlled environments, are undergoing continual change. While adapting to complex lighting changes is trivial for the human visual system, it is a very challenging problem for computer vision system. Changes in the environment's lighting, target shadows, and sensor artifacts such as auto-gain correction can change the overall appearance of seemingly "static" scenes. The measured images change significantly, the system must then decide it is not an interesting change. In less controlled environments, outdoors in particular, scenes are much more dynamic and therefore ignoring insignificant changes is much more difficult. Natural motion, such as moving clouds and tree branches pose additional difficulties.

[1]The two most common adjacency measures are the *four-connectedness* and the *eight-connectedness*. In four-connectedness, a particular pixel is considered adjacent only to pixels either directly above, below, to the left or to the right of it. In eight-connectedness, all surrounding pixels sharing either an edge or a corner with a particular pixel are considered neighbors.

When targets are actively trying to avoid detection, systems are required to constantly watch areas that afford trespassers reasonable cover and concealment. By definition, such areas have limited distance visibility with significant occlusion and clutter. Furthermore, targets of interest generally move in a stop-and-go manner and attempt to conceal themselves within the cover, using camouflage to further reduce their visibility. The combined result of limited distance visibility and small target/background differentiation severely limits the usefulness of stop-and-stare approaches using pan-tilt-zoom cameras. Because a missed detection can be, literally, deadly, a systems level approach is required. The properties of each system component must be carefully considered, optimized, and integrated — from sensor optics, to operating system characteristics, to the user interface (UI). As the paper will show, these situations call for a very sensitive system with a very wide field of view – and hence they are a natural application for omni-directional video surveillance and monitoring.

All systems that build the reference model by temporal blending have the problem that targets that are stationary for long periods of time eventually will become part of the reference. When these targets move on, a set of target pixels caused by the *absence* of the targets generate what are often called *ghosts*. For example, cars commonly generate ghost targets in parking lot scenarios.

When targets are sufficiently distant, they generate independent blobs. However, when either targets or their shadows cause occlusion, splitting blobs into independent targets, or regrouping, becomes more difficult. Take the simple example of an office environment in which two people approach each other and shake hands. Two independent blobs may become one, and then separate again. For a system to properly keep a blob per target, higher-level reasoning must somehow be incorporated into the system. Tracking also becomes more difficult as target blobs collide and decompose. If blobs merge and later split, it can be difficult to determine to which original track each blob belongs.

1.3 Paper Overview

This paper discusses the issues related in taking the simple reference model based approach to visual surveillance systems out of the lab and into the woods — developing a state-of-the-art system capable of detecting and tracking small, low contrast and camouflaged targets in complex outdoor settings. For this domain, the detection phase is *crucial*; if targets are not detected tracking is difficult if not impossible. Detection is also an area where the domain constraints make this more difficult than the situations considered in prior work. For example, Figure 2 shows a scene with a sniper in the grass (detected region magnified). Obviously, the camouflage is quite good, but a sensitive motion vision based tracking algorithm with careful background differencing reveals the sniper's location. Frame-to-frame motion is small — a good sniper may crawl at under a tenth of a meter per minute and be motionless for minutes at a time. A video of this sniper being tracked by the Lehigh Omni-directional Tracking System (LOTS)[1, 2] can be found at `http://www.eecs.lehigh.edu/~tboult/TRACK`.

The next section of the paper reviews the domain constraints and analyzes how existing techniques address these constraints. Because camouflaged targets in outdoor scenes are very challenging, we shall see that much of the state-of-the-art does not directly apply. Section 3 discusses techniques for the change detection subsystems. This is followed by a section discussing grouping and presents quasi-connected components (QCC) which is a novel approach of performing spatio-temporal grouping. QCC combines gap filling, thresholding-with-hysteresis and spatio-temporal region merging/cleaning. Then in Section 5, we briefly review the components that are required for a successful Video Surveillance and Monitoring (VSAM) system to operate in the woods. This includes tracking, target geolocation, network communication and user interface issues. Because of the camouflage and occlusion of this domain, LOTS does not address target identification/classification, and therefore it is not discussed in this paper.

While the problem of tracking camouflaged targets is, hopefully, something only a few people will ever have to consider, the challenges this problem presents require substantial advances in surveillance system sensitivity that can be applied in many other domains. It is interesting to note that Foresti, [3], while researching surveillance systems for "varying badly illuminated outdoor environments" devel-

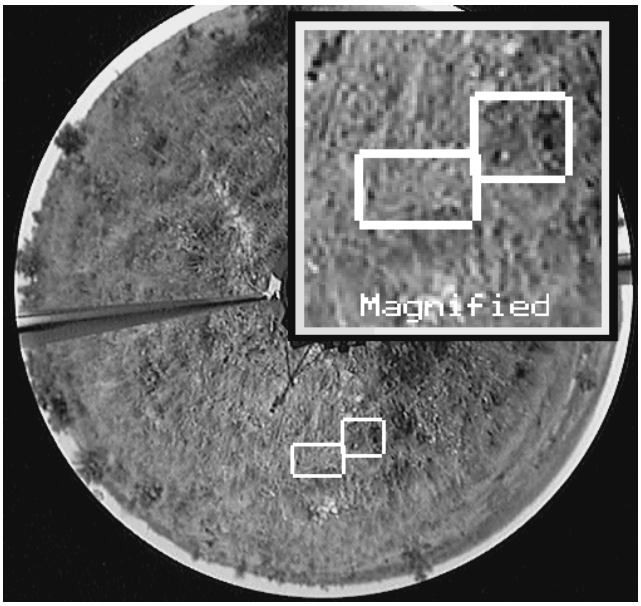

Figure 2: Tracking a sniper moving in the grass – the two boxes approximately cover his upper and lower body (right to left). In a single frame, as is shown here, the sniper is virtually invisible, even though he is only between 3m-5m away.

oped many techniques similar to those presented herein. While there are many differences in the detail, Foresti independently and concurrently, found that surveillance in challenging domains requires change detection combining thresholding-with-hysteresis with a two-level spatial analysis. As we describe our system, we will often come back to compare our approach with that of Foresti.

After reviewing the overall system, we then review our efforts in the analysis of these types of systems and how one can determine the proper system parameters. In particular, in Section 6 we present an error analysis at the pixel and region level that quantifies some of the advantages of QCC. The analysis includes relating the pixel-level errors to region-level errors for both the single-level threshold and the thresholding-with-hysteresis approach. We end with a summary of an external evaluation of the LOTS system performed by the Institute for Defense Analysis and the ongoing applications/experiments within military settings.

The primary contributions of this research are the development of the QCC approach to grouping, the analysis of errors and the approach to parameter setting.

2 Background and Constraints

The primary goal of this paper is to discuss the problem of detecting and tracking potentially adversarial targets in a perimeter security setting, i.e. outdoor operation in moderate to high cover areas. The high clutter and camouflage makes image features difficult to use. We have found only a few other papers within the vision and image processing community that address targets in camouflage, [4], and models of camouflage, [5]. While [4] addresses detection of people in camouflage, it does so by finding a particular simple class of “smooth convex intensity features” that requires thousands of pixels on the target’s non-camouflaged face. The other work, e.g. [5], develops computational models of the signal strength that exists in an image of camouflaged targets, and does not address how to detect targets in camouflage.

This domain of application, low contrast or camouflage targets in high clutter, significantly restricts the techniques that can be applied. Some of the constraints, and their implications include the following.

- Outdoor lighting is naturally and continually varying. The system must be robust enough not to generate false detections caused by sunlight filtered through trees and intermittent cloud cover.
- Trees, brush and clouds all move. While maintaining sensitivity, the system must include algorithms to help distinguish these "insignificant" motions from real target motions.
- Targets need to be detected quickly, when they are still very small and distant, e.g. about 10-20 pixels on target or less than one hundredth of a percent (under 0.01%) of the image.
- Targets use camouflage to blend in, so the system must be very sensitive. Since parts of the target will often match the background, fragmentation is expected. Large amounts of occlusion cause additional fragmentation.
- Many targets will move slowly. Image velocities of under 0.1 pixels per frame are typical with some targets an order of magnitude slower. Some targets will try very hard to blend into the motion of the trees/brush. Therefore, frame-to-frame differencing is of limited value. Furthermore, one must insure that temporal adaption schemes do not cause the blending of slow targets into the background.
- Occlusion, especially in wooded areas is very significant; an average visibility distance in moderate woods is under 50 meters. The directions of targets' motion are only slightly constrained and the entire area must be watched. Combined, these suggest the need for a very wide field of view (FOV).
- Targets consist primarily of humans and occasionally vehicles. Targets will be partially occluded and, in general, will not be "upright" or isolated. Thus, labeling of targets based on simple shape,scale or orientation models is not likely to be successful.
- The algorithms need to be real-time and suitable for use on low cost, low power, embedded Common-off-the-shelf (COTS) systems.

Visual surveillance has been studied for decades with recent major focused efforts in the US, sponsored by DARPA, and Europe sponsored by ESPRIT. The bulk of the prior work has considered indoor or more structured urban settings with relatively large targets having hundreds or thousands of pixels on target, within scenes of medium to high contrast. We very briefly survey some of this existing work and state how the domain constraints impact those approaches. In addition to the papers cited, a good review of many state-of-the-art visual surveillance systems can be found in the August 2000 special issue of the IEEE Transactions on Pattern Analysis and Machine Intelligence dedicated to Video Surveillance as well as recent IEEE Workshops on Visual Surveillance (1998,1999,2000).

There has been considerable work on feature-based, edge-based or boundary-based tracking techniques, e.g [6, 7, 8, 9]. However, for our domain, the targets' small size, deformations and nearly continual partial occlusions limit the applicability of feature-based approaches. Using features to help initialize a stronger model is a powerful tracking technique that has been used by many researchers, e.g. with weak models for people in [10, 11, 12, 13, 14] and strong models for vehicles in [8, 9]. Models permit restricting the search area for likely features, thereby allowing increased sensitivity without significantly increasing the chance for false alarms. However, these systems require both a reasonably large number of pixels on target and model initialization.

The issue of model initialization is even more of a limitation for work on tracking using deformable models, e.g. [6, 7, 15], where the initialization is required to be quite close to the target outline. The deformable models are often far too expensive for serious real-time tracking. For example [7] used 128×128 images and needed 16,000 processors to achieve real-time performance, while [15] needed significant preprocessing per scene and could not handle changing illumination. For some domains, the initialization (and even model tracking) is simplified by the use of color. For example, in [16] and in numerous face tracking systems, skin color is critical to both detection and tracking. For our adversarial targets, color is not likely to contribute to tracking. Furthermore, these algorithms must eventually run 24-7 using thermal or intensified imagery, both of which are monochromatic.

Another class of techniques uses optic flow, but few, if any of these techniques can handle the slow motions and small size of our targets. Many use correlation or sum-of-squared-differences (SSD) over windows [17]. These will not work well with the small targets, large amounts of occlusion and target deformations. Others use feature-based optic flow, computing and tracking features over time, e.g. [8, 18, 19]. The ASSET-2 system, which tracks moving objects against a moving background [8] utilizes a feature-based optic flow. ASSET-2 uses custom hardware and a PowerPC-based image processing system to achieve frame-rate performance. The example tracks provided by the authors are either motor vehicles or aircraft and have many hundreds of pixels on target. More recently, Iketani et.al. explicitly addressed backgrounds that undergo motion [18, 19], with an optic flow based technique that uses a path voting process to detect regions of similar mean flow. It is assumed, however, that the target object's motion can be described with a constant vector. Again large targets with minimal occlusion are implicitly presumed.

There have been many papers on tracking and analyzing human motion, e.g. [10, 11, 12, 13, 14, 20]. Motion parameter analysis has also been used to distinguish targets. For example, [21] uses motion parameters as the primary method to distinguish between human and vehicle. However, it is presumed that targets are not occluded and consist of many hundreds or thousands of pixels. This limits its applicability in our domain. In [10], a system is presented that uses both motion parameters and target size/shape information to classify targets as human, bird, rabbit, fox or squirrel. The paper mentions small (25 pixel) targets, but uses size for classification, resulting in such small targets being classified as birds. Other related research has worked on developing target motion estimators, e.g. [22, 23, 24, 25]. These, however, are focused more on estimating models (usually smooth, periodic or planar models) of the moving targets. However, areas of cover generally produce apparent target motion that is neither smooth nor planar. The ideas of target identification based on motion patterns might be generalized to apply in our domain. However, the segmentation/tracking processes used in these approaches are insufficient for the complex clutter and outdoor variations inherent to the domain. Their ideas might be applied after more sophisticated detection and frame-to-frame matching and may be useful for further analysis of target type.

Although there has been considerable efforts in the literature in frame-to-frame matching, feature-based techniques, motion estimation, and even target identification, the majority of papers have focused on issues other than change detection. Hence, the detection/grouping techniques of such systems may work well for indoor or simple urban scenes, but are not likely to be sensitive and robust enough for handling camouflaged adversarial targets. To reiterate, the detection phase is *crucial*; undetected targets cannot be tracked. Detection is also an area where the domain constraints make tracking more difficult than in the domains considered in (almost all of the) past papers. As a result, much of this paper (and our system's computational effort) concentrates on the detection phase. Because of the camouflage and occlusion, target identification is not attempted and tracking is limited to matching consistent spatial/temporal motions. However, the sensitive detection/grouping approach presented herein could be used as the first stage in many other domains. If needed, the system parameters can be set to reduce sensitivity.

The final domain constraint to consider is the need for a wide field of view. This usually is accomplished with either multiple cameras or a pan-tilt-zoom camera. While the tracking algorithm presented herein can be applied to a traditional camera, it was developed for use with the omni-directional camera developed by Shree Nayar [26], that uses a single camera and mirrors to capture a full viewing hemisphere or more. This camera produces an image that sees in all directions (e.g. see Figure 2) with an optical system that was designed so that targets could be unwarped into a perspectively correct, normal looking image (see Figure 4.) Considerable work also exists in the area of omni-directional systems with a recent IEEE Workshop in 2000 dedicated to the topic. We have focused on the commercially available paracamera [27] because it permits viewing a very large FOV using only a commercially available small,

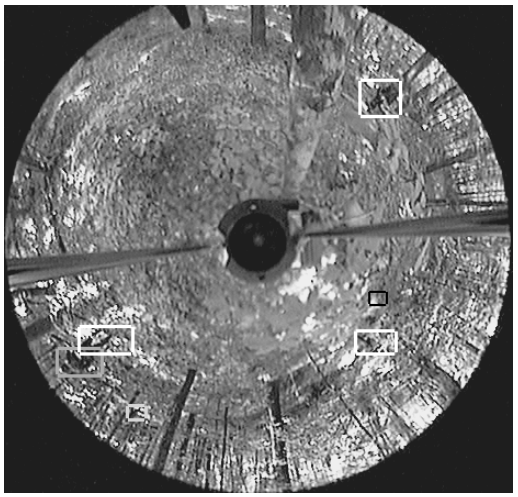

Figure 3: This image shows the tracking of soldiers moving in the woods at Ft. Benning, GA. Each box *is* on a moving target, and only the small white box on the lower left shows a target at significant distance (about 20m). LOTS can detect soldiers at 30m–40m, but this example uses closer targets so the reader can actually see them. :-)

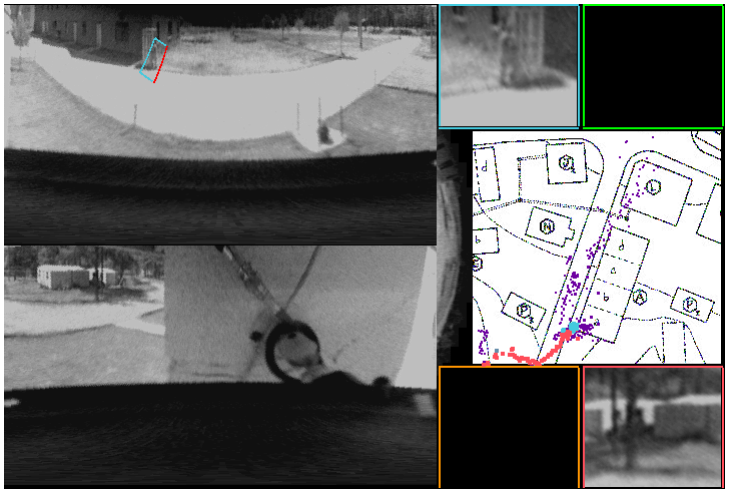

Figure 4: Example showing LOTS interface for the Department of Defense (DoD) Smart Sensor Web program. Left is an unwarping of the paraimage into a pair of panoramic images. The right shows unwarpings of the top four targets, with only two targets in the scene (one entering a building). The map shows the targets' current and recent location history (larger dots are more recent.) Dot color matches the window color showing that target. See Section 5 for more discussion.

single, stationary camera with a single virtual viewpoint. Since a primary goal was the ability to track camouflaged soldiers moving in woods and fields, the omni-directional imaging was a critical feature – in woods, visibility distance is limited, usually to the range 30-50 meters.

It is worth noting that the "spatial resolution" of the paraimage is not uniform. While it may seem counter intuitive, the spatial resolution of the paraimages is *greatest* along the horizon, just where objects are most distant. In [28] we show that along the horizon, the resolution of an omnicamera is 4.2 pixels per horizontal degree, which is about the same as three traditional cameras with 150 degree FOV that would be needed to watch the same region. With either an omni-directional camera or many traditional cameras, objects to be tracked in a wide field of view will cover only a small number of pixels. With 4.2 pixels per degree, a target of dimension 0.5m by 2.0m at 50m is approximately two pixels by eight pixels, yielding 16 pixels per target. At 30m, it is 32 pixels. The numbers stated here presume ideal imaging of the target, while actual imaging, "edge" effects and partial pixel fills reduce the number of effective pixels on target. When one considers that the targets will also be wearing camouflage, as in Figures 2 and 3, it is clear that tracking in such a wide field of view requires the processing of the full resolution (640×480) image with a sensitive, yet robust, algorithm.

In the next two sections, we review in detail the change detection and grouping components of the LOTS system. To illustrate the effectiveness of LOTS, we will present running examples based on some of the most difficult types of change detection — the detection and tracking of a sniper.

3 Change Detection

One of the most common types of change detection is based on subtraction of a background model (or models) followed by thresholding. At the core of this type of change detection is the modeling of an

expected value of a pixel. This section discusses said techniques.

An underlying assumption of many early background modeling approaches was that a single Gaussian would be sufficient to model a pixel value. Since different objects may project to the same image point (if scene points move) and lighting can change, more recent systems provide multiple models, e.g. a Mixture of Gaussians (MOG), per pixel. Existing systems usually set the number of Gaussians, K, within the range 2 to 5 [15, 29]. Furthermore, for computational reasons, the covariance matrix is assumed to be diagonal, i.e. uncorrelated. Obviously, the special case $K = 1$ is the traditional Gaussian model. We also note that, with sufficiently many terms, a MOG can approximate, the case when a single pixel's intensity distribution is *not* well modeled by a single Gaussian.

To use a MOG model, we also need to assume that each underlying data component satisfies a quasi-stationary criterion: the signal is flat fading, i.e. the change in pixel intensity value is slow compared to the update rate of our model. For dynamic MOG models, we also presume the high-level labeling process will correctly indicate which part of the mixture to update. Next, we briefly review previous work on background modeling.

The P-finder system [11] uses a multi-class statistical model for the tracked objects, but the background model is a single Gaussian per pixel. A single Gaussian per pixel, used in many systems, is easy to estimate. If the model is appropriate, then thresholding based on the standard deviation is statistically well justified. Some simpler systems even ignore the formal modeling of standard deviation and simply track the mean or some other models of central tendency and use an ad-hoc thresholding process.

Other papers have stated that the use of a single background can limit robust tracking, especially with outdoor scenes containing significant clutter, e.g. [2, 15, 29, 13], so these systems support multiple background models per pixel. One such model, used in [15, 29], is to fit a MOG to the given input samples. The parametric form of the MOG distributions then can be used to classify pixels. In [2], a simpler form is used that tracks only the central values of the two primary distributions for a pixel. These papers draw mostly on intuition and insight, and do not present experiments justifying their multiple background model assumption or parameter settings.

The PASSWORDS project, [30] uses an illumination change compensation algorithm to allow it to work in outdoor settings. They also employ a shadow analysis to remove shadows using color analysis. They use a background image that is continuously updated to represent the non-moving objects and scenery. Riddler, et.al., [31], uses Kalman filtering for adaptive background estimation which takes into account changing illumination so as not to mistake lighting as objects of interest. They consider the changing velocities of foreground objects so that objects that are temporarily stationary or moving slowly are not blended into the background. A similar approach is used in [3].

There are two approaches for maintaining/updating the background model: multi-sample and per frame processing. A few approaches, e.g. [15, 32], gather many samples per pixel (i.e. many images) and then use the multiple samples to compute statistical models using a MOG and non-parametric models respectively. These methods require considerably more memory and processing and are more complex, e.g. [15] required hours of computation to build its background models, and did not update them as the scene changed.

Per-frame processing approaches seek to compute an updated background model for each new frame. These approaches are probably more common because they require much less storage and much less computation than maintaining $2KN$ images (for a temporal window of size N and K component MOGs). The basic idea is to update the background model via temporal blending (Equation 2). This de facto standard method for background maintenance is examined in the next section. An alternative, which can really be viewed as simply a more principled approach to temporal blending, is to use a Kalman filter, e.g. [3].

In systems with multiple backgrounds, a separate (higher-level) process often determines which of the many backgrounds to update. If true variance estimates are available for each of the many backgrounds,

then the Mahalanobis distance can be used to measure distance of the input from the various background and determine which to use. In LOTS, we do not compute each model's variance, but instead use the straightforward process of simple grey-scale distance to determine which background model is closer.

For this class of reference-image based change detection systems, there are two main components that must be addressed, background modeling and thresholding. We now examine each of these in turn.

3.1 Background Modeling Summary

For the sake of simplicity, we presume a two background model. At some time t, let the primary background be represented by $B_p^t(\phi)$, and the secondary background by $B_s^t(\phi)$. The pixel intensity value is $I^t(\phi)$, where ϕ is the pixel index. For grey-scale images $\phi = (u, v)$ and for n-channel color $\phi = (u, v, c)$. Without loss of generality, we presume the input at time $t-1$ was closest to the primary model $B_p^{t-1}(\phi)$. For performance reasons, if that is not true, we swap the pixels between the two background images to make this likely to be the case in the next time step. We define the difference images to be

$$\begin{aligned} D_p^t(\phi) &= I^t(\phi) - B_p^t(\phi) \\ D_s^t(\phi) &= I^t(\phi) - B_s^t(\phi) \end{aligned} \tag{1}$$

and define variable $q \in \{p, s\}$, as the index with smaller difference D^t and $\bar{q}$ as the remaining index.

In LOTS, background updates depend on feedback from upper layers — updating more slowly in regions we consider to be targets. In particular, we allow for some process to label the pixel ϕ as being in the target set T or in the non-target set N. Then, we can define a generalized update with

$$B_q^{t+1}(\phi) = \begin{array}{ll} [1-\alpha']B_q^t(\phi) + \alpha' I^t(\phi) & \phi \in T^t \\ [1-\alpha]B_q^t(\phi) + \alpha I^t(\phi) & \phi \in N^t \end{array} \tag{2}$$

where α' may be (generally is) smaller than α. The other background model is not updated, i.e.

$$B_{\bar{q}}^{t+1}(\phi) = B_{\bar{q}}^t(\phi) \tag{3}$$

The blending of Equation 2 can serve multiple purposes. Its original motivation was to support temporal changes in lighting. A secondary potential benefit implicitly exploited by many systems but explicitly considered in [14], is that the blending of a moving target with the background produces a "beneficial ghost" of the target's path. The use of $\alpha' < \alpha$ allows the system to more slowly adapt in target regions, limiting how quickly a target will be blended with the background. However, this also results in longer false alarm persistence and limits the value of beneficial ghosting.

If one considers only the natural diurnal changes in lighting, then for most of the day the changes needed to account for this are very small. Nevertheless, many systems, e.g. [11, 33, 34, 21], use a considerably large α. This larger value may be explained by noting that larger values are better if that is the only mechanism within the system for handling changes caused by fast lighting changes such as moving clouds or targets/specularly induced automatic gain control (AGC) effects. In addition, larger values contribute to beneficial ghosting of targets which tend to fill in gaps within the moving target, thereby increasing the detectability of fast moving targets while reducing sensitivity for low contrast and slow moving targets. Later, we discuss how LOTS handles these issues by using multiple backgrounds and a separate lighting change detection algorithm.

An implementation issue of using a model similar to Equation 2 is that it generally requires double precision images, especially with small update values. As discussed in [2], using very small blending parameters while using only integer images and integer math requires some tradeoffs. For the sake of both speed and maintenance of numerical accuracy, LOTS does not update the background images every frame. Instead, it reduces the rate at which the background is updated such that the multiplicative blending factor was at least 1/32. For example, an effective integration factor with $\alpha = 0.0000610351$ is achieved by adding in $\frac{1}{16}$ of the new frame to the background every 1024^{th} frame. This slower approach has a secondary advantage of reducing cost. Analysis of LOTS showed that, if the background is updated

each frame, it became the most computationally expensive component of the system, larger by a factor of 4-6 than the next most expensive operation of subtraction and thresholding. Because of this, with its usual settings, the system only calls the update process once each 64 to 512 frames. Since an update requires about one million operations (two multiplications, an add, and a shift per pixel), this produces a savings of 60–480 MIPS.

Further analysis, motivated in part by our analysis of [35], discovered a minor difficulty with this approach. The analysis suggested that very small values of α are most beneficial. However, with the use of integer images and updates, even if the fraction is 1/16, the update rule cannot reduce the difference to zero because the final few bits are never affected. Thus, for fast implementations, we developed a new update rule that we call the *up-down* or the conditional increment model:

$$B_q^{t+1}(\phi) = \begin{cases} B_q^t(\phi) - \eta & \text{if} B_q^t(\phi) > I^t \\ B_q^t(\phi) + \eta & \text{if} B_q^t(\phi) < I^t \\ B_q^t(\phi) & \text{otherwise} \end{cases} \quad (4)$$

where η is the update parameter. Again, one could use floating point arithmetic and allow arbitrary η, but we stick to integers and implement fractional $\eta < 1$ using temporal sampling.

Per application, the up-down model requires less computation and allows the background to exactly match the input with just 8-bit integer math which permits MMX optimizations.[2] The most important drawback to the conditional increment model is that if a target is mislabeled, the system does not "blend" it in significantly; it does not matter if the target is nearly the same or distant in grey values, the update is constant. Since the primary goal of this aspect of the system is to update the background to handle diurnal lighting changes (which should be slow), using a scaled difference does not seem justified. Rather than always blending quickly, LOTS has a separate rapid lighting change detection subsystem that temporarily changes the system behavior when large (i.e. non-diurnal) lighting changes occur. When strong lighting changes are detected, the current system temporarily increases its thresholds while also switching to a larger α-blending-based algorithm to more quickly adapt away the changes. The switch between the modes is automatic and based on rate of growth in a number of pixels labeled as targets. If the growth rate is very radical, such as might occur if an adversary shines a laser directly into the imaging system, the system reports it immediately and tries again on the next frame. With this separate lighting model change technique (also suggested in [36]) the system can maintain high sensitivity while maintaining robustness.

While the majority of existing systems of which we are aware use the blending with a multiplicative factor, there is one other paper that uses an additive update rule. In [3], which addresses surveillance in a badly illuminated environment, a Kalman filter is used for background model update. That filter results in an additive factor to the current background model, with the factor composed of two terms, one for slowly varying illumination and one for white noise. Since neither of these terms should vary quickly, the additive term from that Kalman filter will generally be zero or ± 1.

Unlike [3], LOTS uses multiple background models which allow it to better handle complex background clutter including objects such as trees and grass that move but whose motion is considered insignificant. Figure 5 shows an example of the difference background images. The trees in the scene, and the pine needles visible on the lower left of the image move significantly during the training exercise. The primary background image is visually indistinguishable from the input image shown on the left. The right image shows the secondary background. In this image, the white pixels are those that never required a second background (and hence do not have one). The non-white pixels show the secondary pixel value at that location, e.g. the darker pixels in the neighborhood of the tree on the upper right and the pine needles in the lower left are noticeable. Note that the single pine needle is responsible for a significantly larger region in the second background image because it has a larger range of motions. LOTS

[2] It requires only a compare and addition, blending requires at least 2 add/subtracts and 2 multiplies, and cannot be done in 8-bit math.

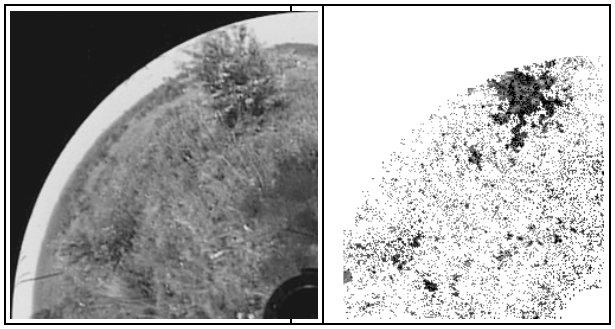

Figure 5: The left image shows a section of a paraimage containing a sniper that illustrates the multiple backgrounds used in LOTS. Left is the primary background, right the secondary background.

always keeps the "closer" pixels in the primary background. Hence, if in the next frame the needle moved upward, the darker pixels at its current location in the primary background would no longer match and the associated pixels in the primary and secondary images would be swapped. In this sense, the moving background may appear to move in the primary background image as well.

3.2 Thresholding

Given the background model, the change detection subsystem still needs to decide if a pixel's change is significant. A classical approach, presuming a Gaussian or MOG model for the background B is to compute the mean $\mu(B)$ and variance $\sigma(B)$ and use standard statistical tests for the thresholding. For example, we label a pixel ϕ according to

$$T^t(\phi) = \begin{cases} 1 & \text{if } |\mu(B(\phi)) - I^t(\phi)| > 2\sigma(B(\phi)) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where a two σ test would give us a detection rate of 95.1%. Of course, any other multiple of σ could be used with different choices on the miss detection/false alarm rates. In LOTS, we do not presume a Gaussian model and our thresholds are based on the dynamic model discussed.

While statistically sound, maintaining a true variance model is expensive and only appropriate if noise is an additive stationary Gaussian. This *static* modeling is discussed in detail in [35] where it is shown that for this static analysis, a single Gaussian, rather than a MOG, generally has a 15% to 200% larger error. The intuitive reason is that the MOG can account for various non-Gaussian features in the distribution.

In Figure 6, each picture shows the histogram of the pixel intensity value of one pixel when the time is changing. Each row in each picture represents a histogram from 100 samples in consecutive time intervals. The vertical axis of these graphs represents time. The darker the points in these histograms, the higher the counts are. Figure $6(A)$ shows the histogram from a pixel on the ground where the target appears infrequently; $6(B)$ is from a pixel on the ground where the target appears more frequently; $6(C)$ is from a pixel of a swaying leaf; $6(D)$ is a pixel on a waving short-grass area; $6(E)$ is a pixel on the wall of a building near the parking lots and $6(F)$ comes from shadows in the parking lot. Only pictures $6(A)$ and $6(B)$ contain any targets. From the figure, we notice that the histogram moves when the illumination changes.

Clearly, these figures suggest non-stationary distributions. Therefore, using variance for thresholding is not appropriate. Because most systems update their background via a process similar to that in Equation 2, these shifts can be removed. However, most of the distributions are also changing in variance — notice the variations in the widths in $6(D)$.

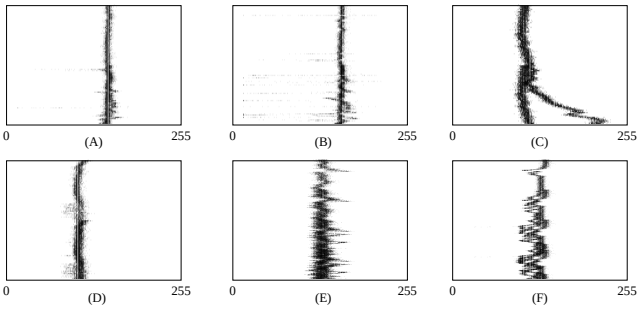

Figure 6: Intensity histograms of different objects over time.

Blending or other low-pass filtering algorithms can be developed for variance-like calculations. For example,

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{6}$$

is used in [29] as a variance update equation based on a new observation, where X_t is the new pixel value. Although such filtering may yield successful dynamic thresholds, the statistical justification of this use of variance is lacking. Furthermore the question of how sensitivity is impacted by this approach has not been explored.

LOTS's approach is somewhat different and intended to maintain high sensitivity. In preliminary system experimentation, we tested a running variance computation but found it expensive and often problematic. The difficulty may be that the underlying noise is non-Gaussian and hence not always well suited to traditional variance tests. For LOTS, we developed an alternative test — see [28] for more details and an analysis. When updating the reference image, the per-pixel threshold is also updated. If the pixel difference from the nearest background is above the per-pixel threshold and leads to a "detected" pixel that did not become part of any region, then the threshold is considered too low and is raised by a constant C_u. If a pixel difference is below threshold, then the threshold is reduced. To increase system stability and reduce false alarms, the threshold increase for noisy pixels is larger than the reduction for below threshold targets. Furthermore, the chance of increasing the sensitivity occurs only $1/C_f$ of the time. (One could also implement this as a fractional reduction of the threshold, but the infrequent update approach allows the use of just integers and is computationally more efficient.) In other words, the approach in LOTS is to replace a comparison with an approximate (measured) variance, which would, if the noise was Gaussian, produce a false alarm a fixed fraction of the time, with a dynamic thresholding that is updated so as to keep the approximate (measured) fraction of false alarms at a constant rate. Rather than fit a model that predicts the false alarms, we directly estimate them and adapt to keep them at the desired rate. Figure 7 shows the dynamic thresholds for a sample scene. The regions where there are two backgrounds tend to have high thresholds because as a pixel changes from its primary color to its secondary, it goes through a range of other colors. This results in fleeting pixel level "detections" that push up threshold. The lower right, where the camera is in the image, has a moderately high threshold because this dark region is less stable due to AGC effects and this higher noise results in a higher dynamic threshold. The image on the right of the figure shows pixels above threshold. Had you noticed the sniper in the grass in Figure 5? He was detected, and those pixels will not be considered false alarms. However, the few isolated pixels (on the left, right and top) will be removed by later processing and hence will cause the dynamic thresholds at those locations to be raised.

The primary disadvantage of this dynamic threshold approach is that it depends on the system's classification of a pixel. Hence, it may improperly adapt a threshold when a very small and low contrast

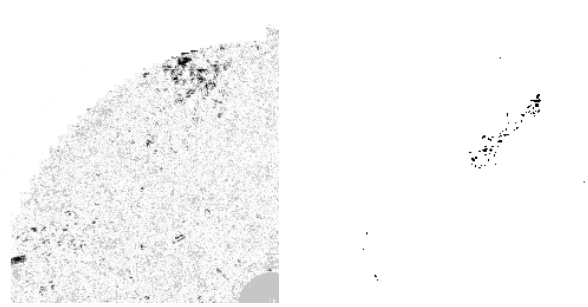

Figure 7: The left shows the system "dynamic" thresholds as an image. Darker pixels represent a higher threshold. The right image shows pixels above the threshold.

target first comes into view, thereby delaying its time to detection. This is the reason we can detect and track regular soldiers at 50m, but we cannot detect the low contrast sniper until about 20m–25m.

In addition to the dynamic per-pixel threshold, the system has a user tunable global "sensitivity" threshold that is a function of the scenario's required false alarm (FA) rate and miss detection (MD) rate. Even with the dynamic threshold and a global threshold, it was still quite difficult to get a good balance between false alarms and missed detections. To address this, LOTS introduced a new approach to thresholding, which is described next.

4 Grouping: Quasi-Connected Components (QCC)

After change detection is applied, most systems form regions by collecting connected pixels. Because there can be small gaps fragmenting the targets, and because there may be small isolated false targets, many systems augment their connected components with morphological processing [34].

This section presents an alternative approach to morphological processing which combines grouping with the thresholding into a process called *quasi-connected components* (QCC). While it would be good to have a detailed comparison of QCC and morphological processing, a comparison would depend very heavily on the image content and parameters used and would be difficult to quantify. Here we simply present the new approach and its analysis in Section 6. One major advantage is that QCC permits that type of probabilistic analysis; similar analysis has proven too difficult to do with morphology.

A main problem for any pixel-level change detection technique is the setting of the threshold for deciding what a "significant" change is. While the analysis in Section 6 provides a principled way of computing a Receiver Operation Characteristic (ROC) curve to make that choice, deriving these ROC curves can be quite labor intensive [35]. However, the tradeoff between the FA and MD rates is often a difficult decision. If one chooses a high threshold to maintain a small FA rate then the MD rate will often soar. On the other hand, the lower threshold needed for a low MD rate would result in a high FA rate. The choice is difficult, even with the knowledge of the ROC curves.

This problem of selecting thresholds is not new. An important approach, that has been very successful in Canny-like edge detectors, is *thresholding-with-hysteresis* (TWH). The idea is to have two thresholds, a high threshold (T_h) and low threshold (T_l). Regions are defined by connected pixels above the low threshold where the region also contains a given fraction of its pixels above the high threshold. In this way, the region has an overall high sensitivity while also trying to insure that at least some of the pixels are very unlikely to be false alarms (since they are above the high threshold). Morphology can fill gaps, but it does so blindly; TWH fills gaps between high-confidence regions in a far more meaningful way.

There are two difficulties here that must be addressed in a TWH implementation. First, an implementation based on iterative region-growing is not efficient enough. Second, even with a low threshold

near or equal zero, gaps will occur because parts of targets, especially camouflaged targets, can match the background exactly. Thus, we still need a technique that can fill across small gaps. Unfortunately, mixing morphology with TWH is not obvious (except perhaps, to apply morphology after region finding with TWH). We propose an alternative approach inspired by our earlier work on G-neighbors, [37]. The approach, which we call *quasi-connected components*, combines TWH with gap filling and connected component labeling. The process efficiently insures that each pixel in a quasi-connected region is "connected" to a given number of pixels above the high threshold, even if the pixel is within a gap.

While we were developing QCC, Foresti [3], was independently developing a system that also uses a thresholding-with-hysteresis based approach. While the details of his implementation are not totally clear, his TWH appears to be quite different. It is unclear if it is a 2D or a 1D threshold-with-hysteresis. The usage discusses local neighborhoods, but it is not clear what information is propagated and how. If it implements region-growing, this may be a part of the reason that the system requires .5 seconds to process a 256×256 image. (LOTS runs at 30fps on a 480×480 paraimage on a 266 MHz PII). It is interesting to note that another researcher working on detection in a difficult environment also found the need for a TWH-like approach.

The processing inherent in QCC is diagrammed in Figure 8. This is a complex figure describing many aspects of a complex process and will be described over the next page of this paper. The description is intermixed with comments on efficient implementation of QCC. Keeping the quasi-connected components process fast is accomplished by three techniques. The most important and interesting efficiency technique comes from a reduction in resolution that simultaneously provides small gap filling. We will discuss this in some detail first. The other two efficiency aids are less significant and will be discussed at the end of this section.

During the detection phase, the system simultaneously builds a lower resolution image of the pixels above threshold, (e.g. the 24×24 image on the top Figure 8 is compressed down to the smaller 6×6 image in the lower left.) Because of its relation to similar concepts in multi-resolution processing, this is generally called the parent image, where each parent pixel has multiple children pixels that contribute to it. The value of each pixel in this parent image is, initially, a count (area) of how many of its associated children (high resolution) pixels were above the low threshold and how many were above the high threshold.

For computational efficiency, one 32-bit integer is used to hold two values — the number of pixels exceeding the low threshold is in the lowest sixteen bits and the number of pixels exceeding the high threshold is in the highest sixteen bits. Because of limited ranges, this allows a single 32-bit addition to combine both counts without the danger of overflow. The shaded pixels in Figure 8 are above the low threshold, the pixels that are both shaded and patterned are above low and high thresholds. Since the resolution is reduced by a factor of four in each direction, either the low order or high order sixteen bits of the parent image pixel storage contains values between zero and sixteen, and allow us to have accurate low-level area counts for thresholding. Connected components are not computed in the full, high-resolution image. Instead, they are computed in the parent (low-resolution) image. As the connectivity is computed, we accumulate the area in terms of high resolution pixels. For example in Figure 8, the difference image shows regions with areas five (upper left grey region), 13 (lower left middle grey region) and 21 (middle right dark region). We note that none of these is completely connected at high resolution, illustrating how QCC can accomplish fine gap filling.

In the bottom left parent image of Figure 8, two different pixels are labeled with their values. The first has value four because it is the parent of four pixels above the low threshold and zero above the high threshold. The second is associated with four pixels above the low threshold, one of which is also above the high threshold. Therefore it has the value 0x0100 + 0x0004 = 0x0401 (65,540 decimal).

To help reject pure noise regions, a low-resolution image pixel with a count of one is ignored when forming the parent image, i.e. before considering connectivity. An example of this is shown on the lower

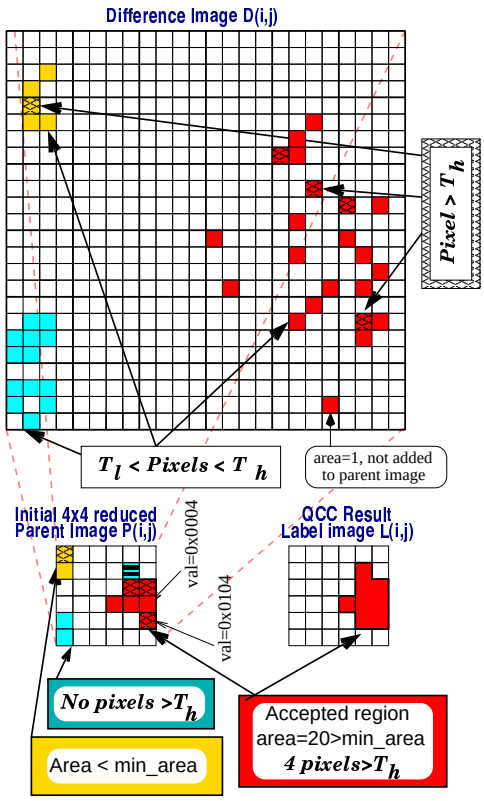

Figure 8: Example showing thresholding-with-hysteresis, quasi-connected components and area thresholding processing.

right of the high-resolution image. In Section 6.2, a detailed analysis of some of the "noise" properties of region-based grouping similar to QCC is provided.

The left image in Figure 9 shows the full "parent" image associated with the frame shown in Figure **??**. The middle shows parent image after running connected components algorithm where targets with sufficient area (6 high resolution pixels) are colored with the region number. The right shows the effect of QCC, where only those regions that can "connect" to pixels above the high threshold remain. Again for the sake of presentation, we are using "huge" targets with more than 300 pixels on target, so the reader can tell which components should be on the target. The system is intended to work with much smaller targets, often with just 10-30 pixels on target. The "target" in this case is the sniper and a small tail of the area of grass that he recently crossed over (which is slightly crushed compared to its original state.)

The setting of low threshold is the sum of the dynamic threshold procedure introduced in the previous section and the global threshold that can be adjusted by the user. The ROC curves described in [35] and summarized later (Section 6.1) are used to set the global threshold depending on the desired MD/FA rate. The desired MD/FA rate is generally a function of the scenario; e.g. for snipers we may be willing to accept a higher FA rate to insure no missed detections.

The high threshold is currently set at a constant either 16 or 32 values higher than the low threshold. This simple added constant above the low threshold has the advantage of being computable by shifting the results using the low threshold. In particular, a pixel is above low threshold is a non-zero value

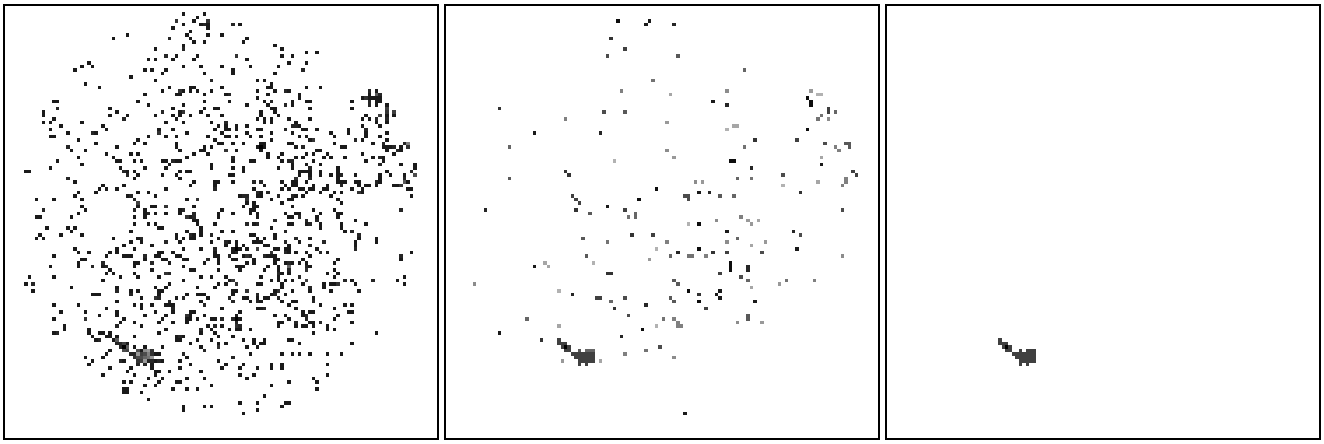

Figure 9: Images showing parent image after various stages of QCC labeling, left is after thresholding, middle after connecting regions, right after full QCC.

results when subtracting, using saturating arithmetic, the low threshold from absolute difference of the new image and the reference image. It is above the high threshold if shifting the low-threshold result is still non-zero. Thus we compute both thresholds in an efficient manner, even with a per-pixel dynamic threshold.

The early version of LOTS simply required a region to have at least one pixel above the high threshold. Because the probability of some noise pixels being above the high threshold increases with the number of pixels in the regions, we recently changed the system to have the number of pixels required to be above high threshold increase to ceil($\frac{1}{128}A$), where A is the high-resolution area of a region.

As mentioned, in addition to the resolution reduction, there are two other techniques used for efficient implementation. First, we use markers generated while thresholding, so that QCC only processes regions between the first and last non-zero parent pixels per row. It also uses markers to simplify processing if the previous row was "empty." These markers are set as we threshold the high-resolution image so that the QCC computations are only computed on segments of each row.

Second, we use a very efficient union-find algorithm [38] — the complexity is nearly linear (with a small constant) in the number of pixels above threshold. The normal connected component phase is only applied to the low-resolution parent image P which produces a label image L. Our `Find` process includes some extra processing to maintain information on which of the new labels match, in a spatio-temporal sense, the previous labels. The `Union` algorithm is standard, except that we extend it to combine the areas of the associated regions. As the regions grow (i.e. as unions occur), we sum the values of the parent pixels within that region. Because of the encoding used, this single addition trivially maintains both the sum of the number of pixels above the high threshold, and the number between the low and high threshold. When the system has computed the region labels, we can decide if a region should be retained. If one wants the image pixels to have the new labels, any connected components implementation requires a second relabeling pass. In QCC, we relabel regions that do not meet the area threshold (either too few pixels overall, or too few above the high threshold) to the background label. QCC may also be extended to include further tests, such as minimal area or region pixel density enforcement to further eliminate noisy regions. In this way, the thresholding-with-hysteresis is essentially computed during the connected component labeling process, without the need for either added thresholding passes or iterative region growing.

The resolution reduction does more than just a data reduction speedup; the resolution reduction also

has the effect of filling in small gaps. However, the gap filling is spatially varying; the maximum distance between "neighbors" varies from four to eight pixels. While LOTS uses a reduction by four, the idea works well with reductions by factors of either two or eight, with more or less the impact on gap filling and region fragmentation that one would expect from larger or smaller windows. While not as "uniform" as morphological processing, it is considerably faster. The advantage, however, is how it naturally combines with TWH. Furthermore, when combined with the area thresholding or density limits, it can distinguish between a "solid" region and a fuzzy collection of isolated points, something morphological processing cannot easily do. While not useful for our applications, higher level morphological processing, such as structured element searching for long-thin targets, still can be applied to the result of QCC processing. We are currently investigating a tighter integration of QCC and morphology.

Given that region detection has been done in the current frame, the system must then attempt to temporally associate current targets with past targets and to analyze the tracked regions. To help handle fragmentation due to occlusions, the "tracking" module in LOTS will also combine two regions that are spatio-temporally close if they are one region in previous frames, if the motion parameters and sizes are consistent and if the system's confidence in the target is high.

Other approaches to grouping include techniques that merge closely related regions. Moscheni, et.al. [39] developed techniques for video coding and robot vision that work only on two consecutive frames. Both spatial and temporal information is used to compute a similarity between regions. They are merged using a weighted directed graph and a graph clustering algorithm. Their paper also contains a good discussion of previous work in spatio-temporal segmentation and merging. Castagno et. al. [40], fuse automatic segmentation with semantic information provided by a user to create segmented video streams. The autonomous segmentation is achieved through an analysis of multiple image features. It is the user's duty to collect the segmented regions into regions of meaning. These approaches might be combined with the simple LOTS merging to increase performance.

5 Tracking within LOTS

This section briefly reviews the remaining components of LOTS — more details can be found in [1, 2, 28]. The tracker runs under Linux using MMX enabled processors. The original system ran using full resolution (640×480) images, at 30fps on a 266 MHz x86 MMX system with 32MB of memory and a PCI frame-grabber. The recent additions, especially the lighting normalized matching and networking interface, reduced the processing speed to 15fps and 12fps respectively on a 300MHz portable/wearable system. MMX instructions are used only for the differencing part of the algorithm. There are many "real-time" tracking systems but the authors are unaware of any others that could, provide sensitive full resolution (640×480) tracking at 15 or 30fps with low-cost COTS hardware. Some of the contributions of this paper are techniques intended to help achieve this type of performance.

Because the noise at each pixel can change, the system maintains a per-pixel threshold. The system adds a global threshold to the per-pixel threshold allowing users to decrease sensitivity. In earlier work, [1, 2], our system was described as having many parameters that were set by hand. While there are many variables in the system, LOTS now has adaptive algorithms that automatically adjust the dynamic threshold, the per-pixel threshold, and the imaging system contrast and brightness. The end user can only choose three parameters — the minimum target area, the global threshold, and the required confidence needed before the system actually reports a target. In practice, we used training data to set the area threshold and global threshold for a variety of scenarios including wooded areas, snipers in fields, soldiers in town, and a mixed wood/field setting.

LOTS uses a two background model similar to that described in Section 3, with two additional background images. The additional images are never blended — they are exact copies of older images. This helps the system ignore "ghosts" that appear when a target enters the scene and persists for 2-5 minutes. More importantly, the system augments the change detection with a lighting analysis. After QCC, each

target is compared using a normalization factor to see if it can be explained as a pure lighting change to that region. Each target region undergoes a normalized comparison (scaling by the average value within the "target") and comparing with the corresponding pixels in the reference image (which are normalized in the same way). Because we have few pixels with target regions, we can afford this more costly analysis which helps to ignore the real, but insignificant, changes caused by moving shadows and lighting changes.

Even with training and proper parameter settings, the system's sensitivity can lead to false detections. To reduce these we use higher level processing, such as the normalized comparisons for lighting changes just mentioned. After QCC and region filtering, the system does temporal association. Most targets are linked in time by QCC directly. However, for those that are disoccluding or strongly fragmented, the system uses the relatively standard idea of matching spatio-temporally nearby targets to maintain a track. This includes searching back many frames to handle small targets occluded by larger obstacles (e.g. closer trees).

While the system needs to be very sensitive, what we choose to report to the user may be only a fraction of the detections. The system computes a confidence measure and by setting the minimal confidence level for reporting, the user can more directly impact the MD/FA rate of reported results. The confidence measure combines the overall target size, its speed in 3D, the quality of the match, occlusion time, its rate of growth (for handling complex lighting false alarms), and its cumulative distance traveled (for handling objects like moving branches before the secondary background model can adapt to include them). The use of cumulative distance traveled is similar in spirit to ideas in [17] though the implementation is significantly different as we do not compute a detailed flow.

Using the image location of a detected target, the system uses the single-viewpoint property of the omni-directional paracamera to back-project that detected target onto a ground plane. System calibration allows the user to specify north, camera height above the ground and its GPS location. Using this, the system back-projects rays to find the 3D position of the targets. On approximately level ground, the system's evaluation is limited by the resolution of the GPS used to gather ground truth — results are often within the 2–3 meters of accuracy in that ground truth.

Three different UI's have been developed for the LOTS system. The two most significant aspects of the interfaces are the geospatial localization of targets and design for efficient bandwidth utilization using the DARPA VSAM protocol [41]. The most recent interface was developed for the DoD Smart Sensor Web program. This interface produces JPEG images (Figure 4) that show the omni-directional image unwarped as a pair of panoramic views. It also shows a map with the targets' 3D positions plotted. The map is color coded so the user can relate targets to one of the four unwarped perspective images and better relate the tracks of targets over time. The map keeps target positions for five minutes so a user can see what regions had activity even if no targets are currently being tracked. The system maintains a database of all these JPEG images and allows users to request tracking results based on time or location.

6 Error Analysis and System Performance

Having looked at the system and some key features of its implementation, we now discuss its performance. The performance is clearly a function of some of the system parameters. In this section, we discuss the setting of those parameters. We begin our discussion of performance at the pixel level using ROC curves, comparing different system parameters and different potential algorithms on a per pixel basis. Then we turn to a more formal analysis of regions.

The first part of the analysis, the pixel level analysis, is relatively independent of LOTS, it does not use QCC but is rather an analysis of the background or reference modeling approach. The region analysis shows the advantages of QCC.

6.1 Pixel Level Analysis and ROCs

The pixel error analysis begins with the computing of the **Probability of False Alarm** (p_{fa}) and the **Probability of Miss Detection** (p_{md}). We convert the p_{fa} and p_{md} into ROC curves which can be used to set system parameters. To produce a ROC plot, all system parameters but one are fixed and a graph of p_{fa} vs p_{md} is plotted as the parameter of interest is varied. One may combine multiple ROC plots for different values of some of the fixed parameters. For background subtraction based systems, the parameters of most significant interest are the thresholds (T_l and T_h) and the blending parameter (α or η).

Receiver Operation Characteristic (ROC) curves/analysis have been used extensively for systems analysis and parameter setting. ROC analysis generally requires considerable experimentation and ground truth evaluation to support the acquisition of the necessary p_{fa} and p_{md} data. A simple, though labor intensive, approach to obtaining p_{fa} and p_{md} is through system operation on controlled data with a labeling of false alarms and miss detections. The desired probabilities then can be obtained from the frequency counts. However, this process must be rerun for every system parameter change, thereby significantly increasing the costs.

It is possible to obtain these probabilities, and hence the ROC curves, more efficiently. In [35], we showed how these variables can be computed from direct measurables. To effectively use this approach we need to set model parameters using real data. Since that is mostly about efficient computation of the ROC curves, it is not presented here. In this analysis we annotated a number of real sequences to get target information and collected background models from even more sequences where annotation was trivial because there were no targets.

The evaluation herein was feasible only because it made heavy use of the equilibrium analysis from [35] which developed models of the system's behavior and derived p_{fa} and p_{md} in terms of simpler measurements. The formulas are a bit long for this paper, but the basic idea is straightforward. Develop a stabilized model of the background using real target free data, then compute p_{fa} and p_{md} by assuming some distribution for the target, and that the targets are sufficiently transitory to not impact the background model state. For systems analysis with ROC curves, this approach allows one to analytically mix different "target" distributions and test them against different backgrounds. By gathering training data on many different inputs, we can have target models for pedestrians, cars, trucks and even targets that try to blend in (i.e. camouflaged targets), and mix them in with different backgrounds. We studied two types of background modeling.

Static Analysis where one solves for the equilibrium state of the background model assuming MOG models for the backgrounds. Once this is done, varying a static threshold is trivial as the entire distribution is known. The equilibrium is recomputed for each desired blending parameter as the blending affects the final distribution. Intuitively, blending tends to shift the different components of the MOG toward each other since any transition from one background to the other may, during the transition, slowly update the distribution with the wrong value.

Dynamic Analysis is needed for the dynamic thresholding approach. For this, one simulates the background/threshold updating to obtain a steady state. One then computes p_{fa} and p_{md} by assuming some distribution for the target, and that the targets are sufficiently transitory to minimally impact the steady state.

Note that since the dynamic analysis depends on the rate of change of lighting, it is computed using multiple training runs. Since these runs are of scenes without targets, the annotation is trivial.

Deriving the equations for the per-pixel probabilities and the equilibrium analysis are beyond the scope of this paper. Since the curves themselves provide insights into the system sensitivity, we present a few examples here. The results here extend the examples in [35] in that they include new results of applying the approach to an alternative background updating using the up-down (conditional increment) model of Equation 4.

In Figure 10 and 11 we present curves that highlight differences in model behavior. In each plot we show various update approaches (parameters in the legend) from Equation 2 and Equation 4. Each curve shows points as the overall detection threshold (T_l) is varied from 0 to 31, — thresholds higher than 31 were not very interesting. These examples are defined using three Gaussians:

- $g_1 = N(127.133, 5.605)$,
- $g_2 = N(132.859, 98.256)$,
- $g_3 = N(72.0128, 159.729)$.

These were determined using the EM algorithm on some of the data used in Figure 6. In the first example (the graphs of Figure 10), we assume g_1 and g_2 are the background distributions and that g_3 is the target distribution, i.e. two backgrounds (one of which is broad), and one target distribution with moderate contrast. The upper graph of Figure 10 is the static case, and lower graph is the dynamic case. The scale on both graphs is $[10^{-4}, 10^{-3}] \times [10^{-3}, 10^{-1}]$. In this "easy case" it is clear that the dynamic modeling is significantly better and that slow updates to the background were better. The blending vs. up-down comparisons are mixed, but the best performances were from the up-down updates. Note that because one of the two backgrounds is quite broad, the p_{fa} is relatively high, and it is easy to incorrectly label one of those random background variations as a target.

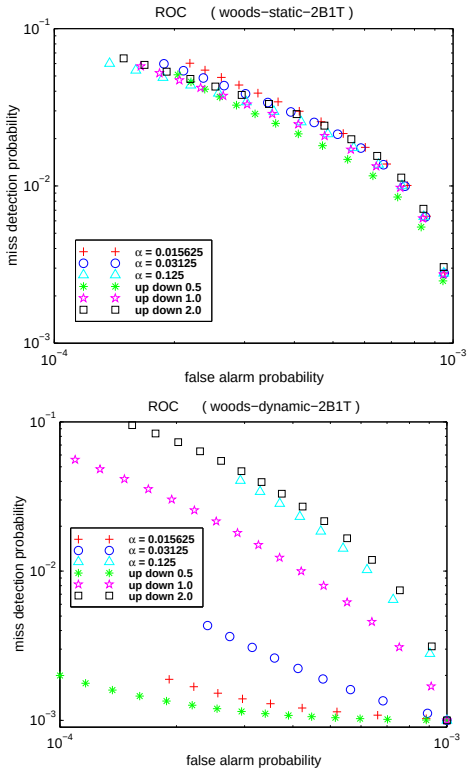

Figure 10: LogLog plots of ROC curves for an easy case, one target and two backgrounds.

In the more difficult case, the graphs of Figure 11, we consider g_1 to be the background and g_2 and g_3 to be the targets. Again, upper graph of Figure 11 is the static case, and lower graph is the dynamic case. The scale on both graphs is $[10^{-7}, 10^{-3}] \times [10^{-3}, 10^{0}]$. In this more difficult case, the dynamic modeling

is better, just not as dramatically as in the simple example. Note how the p_{fa} is now much lower as the background is better modeled, but the p_{md} is increased, especially for larger thresholds. This difficult case models one of the targets being very close to the background, a situation common for most pixels in low contrast or camouflaged targets. Note, however, that even in well camouflaged targets, some pixels at any given point in time, will have high contrast and be more like the easy example. However, these pixels will be sparse and not spatially connected. This is one of the reasons behind the success of QCC.

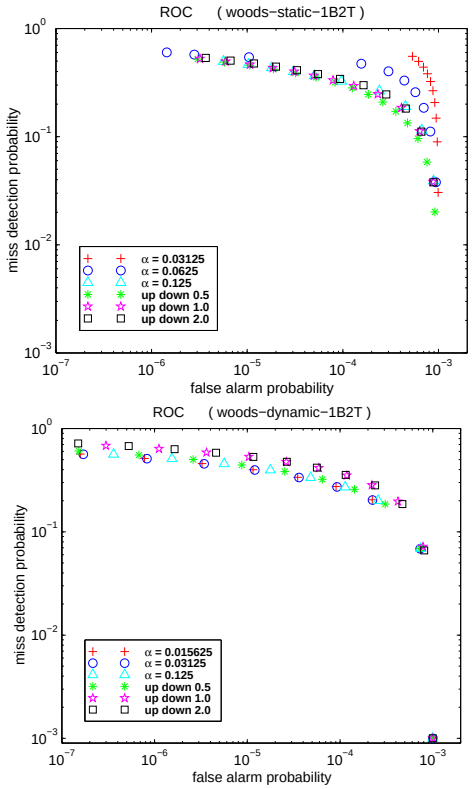

Figure 11: LogLog plots of ROC curves for the difficult example of one background and two targets.

For static modeling, it is clear that the overall performance is much weaker than for the dynamic modeling case. As these curves show, the alternative up-down approach is often superior to the more common blending approach. For the two static case examples, the up-down does better for very low p_{fa}, though the results are mixed for larger p_{fa} values. The up-down approach is markedly better over the whole range of p_{fa} values for the dynamic case, which is the domain in which it was designed to be used.

For a particular domain, using real data and ROC curves similar to these would allow one to determine the appropriate choices for the blending and threshold parameters for a background subtraction based technique.

6.2 Region Level Analysis

Having looked at the pixel level analysis and some example ROC curves that can be used to set parameters, let us now look at how to generalize this analysis to the region level. This is an important generalization of the approach taken in [35] — the region level is where LOTS and many systems begin to distinguish between targets and non-targets. The analysis of the general form of quasi-connected components or regular connected components followed by morphology is, at present, too complex to pursue.

Instead, we consider a slightly simpler model of QCC which is a good approximation for small targets but ignores the non-uniform spatial grouping that would impact larger targets. One of the advantages of QCC is that it permits relative straightforward analysis. Although morphological processing is much older and has a rich mathematical background, the probability analysis to obtain p_{md} and p_{fa} for regions processed with morphology remains elusive.

We develop equations that treat a target region as having a fixed set of r pixels in an underlying region of n pixels.[3] That is, suppose there are n pixels in the whole region and r pixels are associated with a target $(n \geq r)$. Let k_d be the number of pixels that are target pixels that have been detected, hence $(r - k_d)$ is the number of miss detected pixels. Let k_f be the number of pixels that are background pixels that have been incorrectly detected, i.e. the number of false alarms.

We define miss detection probability (p_{rm}) and false alarm probability (p_{rf}) at the region grouping level as follows

$$p_{\text{rm}} \triangleq \sum_{r=0}^{n} p_{\text{md}}(r|RT)p(r|RT) \tag{7}$$

$$p_{\text{rf}} \triangleq \sum_{r=0}^{n} p_{\text{fa}}(r|\bar{RT})p(r|\bar{RT}) \tag{8}$$

where $p(r|RT)$ and $p(r|\bar{RT})$ are the conditional prior distributions on how many target pixels will be in the region given there is a real target in the region, and given there is not a real target in the region, respectively.

For the threshold-with-hysteresis there are two thresholds, called T^H and T^L, with $T^H \geq T^L$. In the following, superscripts show if the parameters are related to the high or low thresholds (they are not powers) and subscripts show if they are related to false alarms (fa, detection (d) or miss detection(md). The term C_n^k represents the combinations n choose k. With this we can derive the joint distribution $p(k_d^h, k_f^h, k_d^l, k_f^l)$ as:

$$\begin{aligned} &p(k_d^h, k_f^h, k_d^l, k_f^l) = \\ &\quad C_r^{k_d^l}(p_{\text{md}}^l)^{m-k_d^l} \\ &\quad \cdot C_{k_d^l}^{k_d^h}(1-p_{\text{md}}^h)^{k_d^h}(p_{\text{md}}^h - p_{\text{md}}^l)^{(k_d^l - k_d^h)} \\ &\quad \cdot C_{n-r}^{k_f^l}(1-p_{\text{fa}}^l)^{n-r-k_f^l} \\ &\quad \cdot C_{k_f^l}^{k_f^h}(p_{\text{fa}}^{nh})^{k_f^h}(p_{\text{fa}}^l - p_{\text{fa}}^h)^{k_f^l - k_f^h} \end{aligned} \tag{9}$$

with the joint distribution of (k^h, k^l) given as

$$p(k^h, k^l) = \sum_{k_d^h=0}^{k^h} \sum_{k_d^l=k_d^h}^{k^l} p(k_d^h, k^h - k_d^h, k_d^l, k^l - k_d^l)$$

where $k^h = k_d^h + k_f^h$ is the number of pixels that are higher than the high threshold, and $k^l = k_d^l + k_f^l$ is the number of pixels that are higher than the low threshold. Obviously, $k^h \leq k^l$.

In QCC, we have two area thresholds called k_m^h and k_m^l that must be satisfied to label a region as a target. $p(k^h \geq k_m^h, k^l \geq k_m^l)$ indicates the probability of how many pixels are higher than k_m^h and how many pixels are higher than k_m^l, where we require $k_m^h \leq k_m^l$.

[3]The mathematics in this section is a summary, the missing steps are not difficult but require a bit of effort to work out. For brevity they are not included.

$$
\begin{aligned}
p_d(r) &\triangleq p(k^h \geq k_m^h, k^l \geq k_m^l \mid r) \\
&= \sum_{k^l=k_m^l}^{n} \sum_{k^h=k_m^h}^{k^l} p(k^h, k^l \mid r) \\
p_m(r) &= 1 - p_d(r) \\
p_f(r) &\triangleq p(k^h \geq k_m^h, k^l \geq k_m^l \mid r) \\
&= \sum_{k^l=k_m^l}^{n} \sum_{k^h=k_m^h}^{k^l} p(k^h, k^l \mid r)
\end{aligned}
\tag{10}
$$

Note that while these equations are defined for thresholding-with-hysteresis, as used in QCC, if one sets the high and low thresholds to the same value, they also apply to single threshold grouping and hence could be used in the analysis of systems with only a single threshold. Also note that we are not enforcing connectivity, thus it is a good model for small targets in QCC, which get lumped into a single parent pixel or a few adjacent ones but not for very large but sparse regions. For this reason we only consider moderately small targets in the remaining discussion.

Using the above equations and data from the individual pixel ROC analysis presented in the previous sections, one can generate ROC curves for regions. This does not include the spatial analysis of QCC, but begins to show how some of the system parameters, including the minimum area size and the dual thresholds play a role in determining the MD/FA rates. In Figure 12 we consider the region analysis building on the pixel level results that were presented in in Figure 11, i.e. for a difficult case with one background g_1 and two targets g_2, g_3, where g_2 is very close to g_1. The graphs consider four system parameters: the low threshold, the high threshold, the minimum low threshold region size KmL, and the minimum high threshold region size KmH. The target size is modeled as a Gaussian distribution with $\mu = 12, \sigma = 2$. We use frequency data for targets from real data, where targets occurred in approximately 0.002% of the frames. The top graph of Figure 12 shows curves where a single threshold is varied within the curve and each curve is a different region size KmL. Note the scales are radically different from the per-pixel case—$[10^{-120}, 10^{0}] \times [10^{-19}, 10^{-2}]$. The second graph of Figure 12 shows curves with the low threshold set at 2, and the high threshold varying (from right to left) from 2 to 18. When the high threshold is equal to low threshold, the right most point on each curve, it shows the non-hysteresis case or single threshold case. Different curves show a different number of points required above the high threshold. Increasing the required number of pixels above the high threshold is better. The third graph of Figure 12 shows the interaction of the number of pixels required above each threshold: within a curve the number of pixels above the low threshold varies (from right to left) from 2 to 14; each curve has 1, 2, 3 or 4 pixels above the high threshold. Overall, for this difficult case we see that threshold-with-hysteresis has significant added value — it allows orders of magnitude reduction on false-alarm rates with only a minor change in the miss detection rates. For the easy case, i.e. building from Figure 10, the system was already doing well and added benefit of TWH is measurable but not as significant.

7 External System Evaluation

To support the evaluation of LOTS, data was collected using omni-directional sensors at Ft. Benning in scenarios of interest to the DARPA Small Unit Operations — Situational Awareness System (SUO-SAS) program. Approximately 70 hours of omni-directional video was collected in the first evaluation and another 40 hours in the second. Both sets include both significant amounts of "targets" and empty scenes for false alarm evaluation. Atmospheric conditions range from light rain and wind, partly sunny and windy to sunny with light breeze. We note that in many of these scenarios it is very difficult to detect and

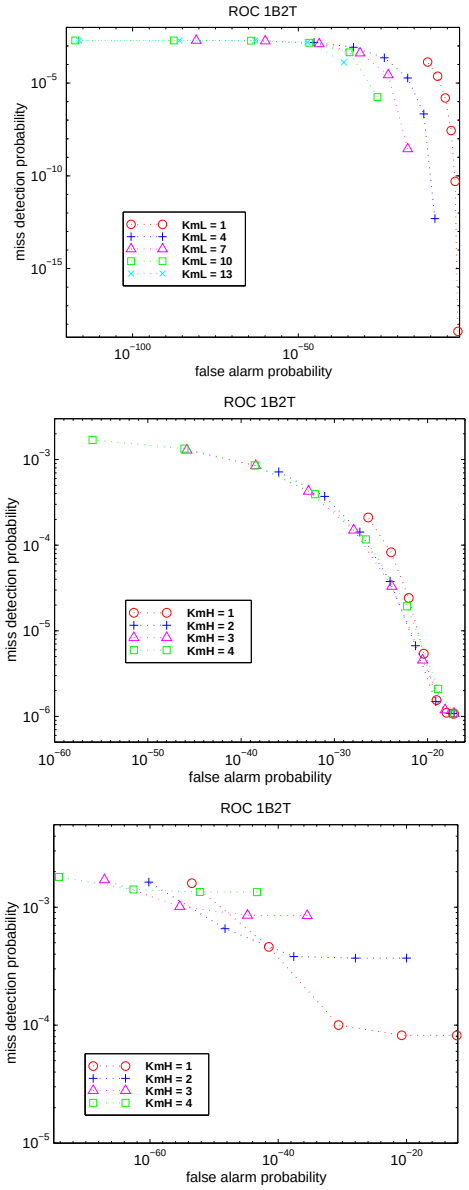

Figure 12: LogLog plots of region ROC curves with respect to different system parameters.

track targets, as can be seen in the two examples of Figures 2 and 3. Readers can find video examples [4] of the tracker in action as well as raw data for testing at

`http://www.eecs.lehigh.edu/~tboult/TRACK/`

Evaluation of this type of system is non-trivial and somewhat subjective. When there is significant occlusion and camouflaged targets, it is often hard to say if a target should be visible or not. It is also

[4]Note for effective transmission on the web the "results" are MPEG files which means they have lost a small amount of image quality.

	Certainty > 0		Certainty > 1	
Scene type	% Detect	p_{fa}	% Detect	p_{fa}
Indoor 1	100%	1.00E^{-4}	100%	0*
Intersection 1	89%	1.83E^{-4}	89%	0*
Intersection 2	87%	4.61E^{-4}	62%	0*
Town Edge	95%	5.00E^{-4}	92%	1.89E^{-4}
Forest	100%	3.33E^{-4}	76%	0*
Field (sniper)	100%	5.89E^{-4}	82%	5.56E^{-5}
Mean	95%	3.61E^{-4}	84%	4.07E^{-5}
Std.Dev.	6%	1.91E^{-4}	13%	7.59E^{-5}

Table 1: False alarm and Miss detection rate (per frame) of basic LOTS tracker as of Aug. 1998. False alarms are per frame, detection rate is the fraction of all targets. Across the scenarios the number of targets ranged from 8 to 30, and in all but the indoor settings the targets were generally at a distance of 20-50m (80-12 pixels on person). This is before lighting algorithms and changes to background modeling and without adaptive parameter adjustments. Main sources of false alarms were about 60% insignificant motions (e.g. leaves and bugs), 30% lighting & shadows. *A miss detection or false alarm rate of 0 resulted because in the approximately 15000 frames per scenario that were evaluated, that type of event was not found.

not clear when something is a false alarm — e.g., take the ambiguous cases of animals, insects, or the emergence of a new motion pattern of brush that might be worth investigating. Rather than presenting our own evaluation, we report on an external analysis of LOTS, as of August 1998. This evaluation, [42], was done by researchers at the Institute for Defense Analysis (IDA), where their goals were to see how well video surveillance and monitoring could be used to support small unit operations.

The 1998 scenarios included a short indoor segment, two urban/street (intersection) scenes, a town perimeter (town edge and a nearby tree-line), two different forest settings, and a sniper in a grass field. For the forest and field scenes, the evaluation was limited to a 2–4 minute batch learning phase for acquiring the multiple-backgrounds, while the others had at most 30 seconds of learning. No learning based on user feedback of false alarms was allowed, though it is supported by the system.

The summary analysis is shown in Table 1. Almost all detections were considered “immediate,” with only the most difficult cases taking longer than one second. The average number of frames evaluated per scenario was approximately 15,000 (approximately eight minutes.) False alarm rates are presented here per frame while the original report used false alarms per minute.

We point out that the evaluators originally labeled many detections as false alarms until they more carefully analyzed the video and data logs and found they had missed targets the system had detected. For example, all rectangles in Figure 3 are true detections, but this may be difficult to tell from the image because some of the targets are small and of low contrast. In the forests and field scenes, most of the miss detections were targets with low contrast moving in areas where there was often ancillary motions (i.e. where the system had multiple backgrounds and therefore reduced sensitivity). In the intersection scenes, most of the missed targets were either too small (but with enough contrast that the human could see them), or they were in areas with ancillary motion and multiple backgrounds. The main false alarms in the town scenes were complex lighting/shadow effects while animals, bugs and some branches were dominant false alarms in the forest and field scenes.

The IDA evaluations did not include any of LOTS’s reported confidence measures (they were in the output, but not considered). We took the detailed spreadsheet from their report, which showed in which frames targets were detected, and then went back to the system output and included the confidence values to produce the second set of columns in Table 1. The computations with confidence levels were done at Lehigh (not by the independent evaluators), but were based on their scoring of what was a false alarm

and what was a true detect. We also used the exact same system output video tapes produced for the external evaluation.

The initial evaluation and analysis did not allow for any incremental learning nor adaptive feedback on false alarms. Incremental learning is intended to handle fast changes in lighting. Once the end-user said an effect was a false alarm, the secondary background could account for the lighting. Without that feature, a large fraction of detected false alarms were small to moderate sized locations with lighting related changes, (e.g. small sun patches or shadows.) In a wide field of view, many of these lighting effects can produce image regions that look like a person emerging from occlusion or a moving low-contrast vehicle, which is why we intended to use user feedback to initially label them as false alarms. The "ghosting" of targets was also noted in their report, and they too were considered false alarms. The system needed to be more automatic because military use cannot support that level of user feedback. This requirement led to additional cleaning phases, in particular the introduction of lighting algorithms and the use of the old-image approach to handle mid-term ghosting. Our updated system is a component of SUO-SAS program (in a project lead by CMU) that has been delivered for long-term evaluation at Ft. Benning. The new version also includes 3D target localization as seen in Figure 4. The new series of evaluation includes multiple camera configurations and determination of both localization accuracy and detection/false alarm rates. The preliminary data analysis from one camera showed a (still unofficial) localization of within 2m and a decrease in the false alarm rate. Formal results are expected to be released in 2001.

This latest version of LOTS is being used as a component in the DoD Smart Sensor Web program where it is currently being applied in a more urban setting. Development of an 8–14 micron infra-red version of the system is currently underway.

8 Conclusions

Detection and tracking of camouflaged targets requires both sensitivity and robustness. This paper show how we have taken such systems out of the lab and into the woods. It presented an overview of the LOTS system that has demonstrated the ability to track these targets and described some of its unique design choices.

The major contributions include a new approach to grouping called quasi-connected components, which was introduced in Section 4. QCC implements a two level threshold-with-hysteresis approach that fills very small gaps even if there is no connection, and fills larger gaps if there is a bridge of pixels above the low threshold connecting them to something above the high threshold. This approach has been used in edge detectors and provides a unique and efficient approach for its use in region detection for visual surveillance systems. The implementation presented is significantly faster than previous 2D region growing approaches.

The paper discussed the advantages of a multiple-background approach, which has been used by others, but with the novel features of a new conditional increment background modeling for very slow updates (Section 3.1), the additions of the non-blended background images that handle "ghosts" and lighting change detection algorithms (Section 5).

Section 6 presented a discussion on the error analysis of the region detection and use of ROC curves to help understand the performance of and determine parameters for the change detection subsystem. It presented data showing how miss detection and false alarms rates vary at the pixel level, and compared the well known blending approach with the proposed conditional increment background modeling approach. We then showed how, as an approximation to QCC for small regions, the pixel level analysis could be extended into a region level analysis. The ROC curves generated from this new error analysis clearly show the advantages of thresholding-with-hysteresis for difficult visual surveillance problems. This theoretical analysis confirms the advantages of QCC that had been observed in practice, it allows orders of magnitude reduction on false-alarm rates with only a minor change in the miss detection rates.

The paper then discussed the overall performance of the system, as measured by an external evaluation group. While the paper has shown these techniques in the context of low contrast and camouflaged targets, the external evaluations show that these ideas can be applied to other less demanding domains. While the system represents a major advancement, there are many challenges remaining in this domain including: better techniques for distinguishing significant motions from real but non-interesting motions, target identification, better maintenance of target identity over occlusions, full 24 hours per day, 7 days per week operation, and multi-sensor fusion.

Our work, as well as that of Foresti ([**?**], suggests that for visual surveillance in domains with low contrast targets moving in changing environments with high occlusion, we can conclude with thresholding with hysteresis and multi-level analysis play a major role in the development of effective solutions. Given that quasi-connected components is not only effective, but computationally inexpensive, we expect techniques like QCC will become a major component of future visual surveillance systems.

References

[1] T. Boult, A. Erkin, P. Lewis, R. Micheals, C. Power, C. Qian, and W. Yin, "Frame-rate multi-body tracking for surveillance," in *Proc. of the DARPA IUW*, 1998.

[2] T.E.Boult, R.Micheals, X.Gao, P.Lewis, C.Power, W.Yin, and A.Erkan, "Frame-rate omnidirectional surveillance and tracking of camouflaged and occluded targets," in *Second IEEE International Workshop on Visual Surveillance*, pp. 48–55, IEEE, 1999.

[3] G. Foresti, "Object detection and tracking in time-varying and badly illuminated outdoor environments," *SPIE Journal of Optical Engineering*, vol. 37, no. 9, pp. 2550–2564, 1998.

[4] A. Tankus and Y. Yeshurun, "Detection of regions of interest and camouflage breaking by direct convexity estimation," in *First IEEE International Workshop on Visual Surveillance*, pp. 42–48, IEEE, 1998.

[5] A. Copeland and M. Trivedi, "Signature strength metrics for camouflaged targets corresponding to human perceptual cues," *SPIE Journal of Optical Engineering*, vol. 37, no. 2, pp. 582–591, 1998.

[6] D. Huttenlocher, P. Noh, J. Jae, and J. William, "Tracking non-rigid objects in complex scenes," in *International Conference on Computer Vision*, (Berlin), Sept. 1995.

[7] J. Woodfill and R. Zabih, "An algorithm for real-time tracking of non-rigid objects," in *Proc. of the National Conf. on AI*, pp. 718–723, July 1991.

[8] S. Smith and J. Brady, "ASSET-2: Real-time motion segmentation and shape tracking," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 814–820, 1995. Similar material in *Eng. Apps. of AI* April 1994, and (without Brady) in ICCV95.

[9] D. Koller, K. Danilidis, and H. Nagel, "Model-based obect tracking in monocular image sequences of road traffic scenes," *International Journal of Computer Vision*, vol. 10, no. 3, pp. 257–281, 1993.

[10] P. Rosin and T. Ellis, "Detecting and classifying intruders in image sequences," in *Proc. of British Machine Vision Conference*, pp. 293–300, Sept. 1991.

[11] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[12] R. Polana and R. Nelson, "Low level recognition of human motion," in *Workshop on Non-rigid Motion*, pp. 77–82, Nov. 1994.

[13] I. Haritaoglu, D. Harwood, and L. Davis, "W^4s: A real-time system for detecting and tracking people in 2.5D," in *Computer Vision—ECCV*, 1998.

[14] J. Davis and A. Bobick, "The representation and recognition of human movements using temporal templates.," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 928–934, 1997.

[15] S. Rowe and A. Blake, "Statistical background modelling for tracking with a virtual camera," in *Proc. of British Machine Vision Conference*, 1995. Web version of a similar TR also availble.

[16] T. S. J. Weaver and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 1999. See Also MIT Medial Lab TR 466.

[17] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, pp. 774–781, August 2000.

[18] A. Iketani, Y. Kuno, N. Shimada, and Y. Shirai, "Real-time surveillance system detecting persons in complex scenes," in *Proceedings of IEEE International Conference on Image Analysis and Processing*, pp. 1112–1115, IEEE, 1998.

[19] A. Iketani, A. Nagai, Y. Kuno, and Y. Shirai, "Detecting persons on changing background," in *Proceedings of IEEE International Conference on Pattern Recognition*, pp. 74–76, IEEE, 1998.

[20] I. Haritaoglu, D. Harwood, and L. Davis, "W^4: Real-time surveillance of people and their activities," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, pp. 809–830, August 2000.

[21] A. Lipton, H. Fuijiyoshi, and R. Patil, "Moving target detection and classification from real-time video," in *Proc. of the IEEE Workshop on Applications of Computer Vision*, 1998.

[22] S. Blostein and T. Huang, "Detecting small moving objects in image sequences using sequential hypothesis testing," *IEEE Trans. Signal. Processing*, vol. 39, no. 7, pp. 1611–1629, 1991.

[23] S. Ayer, P. Schroeter, and J. Bigun, "Segmentation of moving objects by robust motion parameter estimation over multiple frames," in *Computer Vision—ECCV*, vol. Vol. 2, (Stockholm), pp. 316–327, May 1994.

[24] R. Cutler and L. Davis, "Robust real-time periodic motion detection, analysis and applications," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, pp. 781–796, August 2000.

[25] Y.Ricquebourg and P. Bouthemy, "Real-time tracking of moving persons by explorting spatio-temporal image slices," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, pp. 797–808, August 2000.

[26] S. K. Nayar, "Catadioptric omnidirectional camera," in *Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 482–488, July 1997.

[27] "Remote Reality, Inc.." Makers of ParaCamera systems, `www.remotereality.com` formally known as cyclovision.

[28] T.E.Boult, R.Micheals, X.Gao, A.Erkan, W.Yin, and C.Power, "Omni-directional frame-rate detection and tracking of camouflaged and occluded targets," tech. rep., Lehigh, Sept. 1999. Submitted.

[29] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246–252, IEEE, 1999.

[30] M. Bogaert, N. Chleq, P. Cornez, C. Regazzoni, A. Teschioni, and M. Thonnat, "The passwords project," in *ICIP*, pp. 1112–1115, IEEE, 1996.

[31] C. Riddler, O. Munkelt, and H. Kirchner, "Adaptive background estimation and foreground detection using kalman filtering," in *ICRAM*, pp. 193–199, 1995.

[32] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *FRAME-RATE Workshop*, IEEE, 1999. Eletronic (only) proceedings at www.eecs.lehigh.edu/FRAME.

[33] W. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using adaptive tracking to classify and monitor activities in a site," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 22–29, 1998.

[34] B. Flinchbaugh and T. Olson, "Autonomous video surveillance," in *25th AIPR Workshop: Emerging Applications of Computer Vision*, May 1996. See also DARPA IUW May 1997.

[35] X. Gao, T. Boult, F. Coetzee, and V. Ramesh, "Error analysis of background adaption," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 2000.

[36] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *International Conference on Computer Vision*, pp. 255–261, IEEE, 1999.

[37] T. Boult, R. Melter, F. Skorina, and I. Stojmenovic, "Applications of g-neighbors to image processing and morphology," *Machine Graphics and Vision*, Nov. 1996.

[38] R. Sedgwick, *Algorithms in C++*. Addison-Wesley, 1992.

[39] F. Moscheni, S. Bhattacharjee, and M. Kunt, "Spatiotemporal segmentation based on region merging," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 9, pp. 897–915, 1998.

[40] R. Castagno, T. Ebrahimi, and M. Kunt, "Video segmentation based on multiple features for interactive multimedia applications," *IEEE Tran. on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 562–571, 1998.

[41] A. Lipton, T. Boult, and Y. Lee, "Video surveillance and monitoring communication specification document 98-2.2," tech. rep., CMU, Sept. 1998. `http://www.cs.cmu.edu/~vsam/Documents` as `vsam_protocol_98_22.ps.gz`.

[42] C. Dion-Schwarz and J. Silk, "Evaluation of vsam potential for suo-sas." Presentation at 1999 DARPA VSAM Workshop, Pittsburg. Contact cdion@ida.org for copies, Oct. 1998.

Changes

Here is a short list of the changes made (directly tied to comments by reviewers).

- (R1) organization of the paper could be improved (we added the tutorial stuff to the Introduction and reduced the abstract etc.)
- (R1) Introduction and Background sections could be shortened (based on other referee's comments we needed to add material to the intro – Background was shortened)
- (R1) Titles of figures are too long (addressed)
- (R1) Evaluation section could be more emphasized in the paper to show results and more discussion of field test (would've significantly increased length of paper, so it was not done.)
- (R2) possibly not enough introductory material for the non-specialist (tutorial section in Intro addresses this.)
- (R2) paper relies too heavily on [29] Gao Boult Coetzee and Ramesh – should make the paper as independent of [29] as possible (was made more independent.)
- (R2) need more details after: More importantly, the system augments the change detection with a lighting analysis: (addressed there.)
- (R2) figures 11-13 legends too small (addressed)
- (R2) brief qualitative analysis of the techniques of *general* applicability (see conclusion)
- (R3) no comparison of grouping with existing methods (comparison with morphology scattered throughout the paper.)
- (R3) clearer claims in the abstract and intro (addressed)
- (R3) removing some figures and making the remaining ones more understandable (addressed)
- (R3) polishing the English (done)
- (R3) Major modifications to mathematical formulations and explanation of results should be clearer and easier to read (see later clarifications)
- (R3) Conclusions are far too brief – need much more detail – possibly including new subsections referring to various claims
- (R3) all figure captions are too long (addressed)
- (R3) There is no prior reference to the threshold parameter G which is discussed here: The two parameters of most significant interest are the threshold G and (addressed)
- (R3) page 5: computational cost for windowed mean and variance are given for granted and should be explained (no computational cost is given, only storage cost.)
- (R3) page 5: greyscale distance vs. more standard Mahalanobis distance should be justified (comments added)
- (R3) page 5: need to cite references for this sentence: In some systems, including the early implementation of LOTS, background updates depended on feedback from upper layers. (deleted)
- (R3) page 6: The claim that the "up-down model requires less computation" is not justified (we thought it was obvious that up-down-model uses if test and add, the other requires a subtr, two mults and an add)
- (R3) In the Thresholding subsection, equation (5) on page 6 – may need a modulo on lhs (added absolute value)

- (R3) page 7: equation 7 of [25] needs to be expanded (include the equation or expand the explanation) (the equation was added and explained)
- (R3) page 7: MOG background models are more of a convenience ... not justified (removed sentences)
- (R3) page 8: The figure (9) inadvertantly appeared before figure 8, disassociating it from the description. Description modified to address this.
- (R3) page 10: (added a higher level processing example to the sentence)
- (R3) page 11: comment added to highlight geospatial localization – section 6 rewritten –
- (R3) page 12: figure 11 simply announced (corrected added to text)
- (R3) page 12: The sentence: In each ROC curve, the global threshold was varied from 0 to 31, although ... should explain why 0 to 31 was used. (thresholds greater than 31 are generally off the graphs.)
- (R3) page 13: formulas (6) through (12) hard to understand (addressed)
- (R3) usual marathon captions – It's a question of style. We believe figure captions should be stand alone and not say "see text for discussion." Since multiple referees suggested the change we have capitulated.
- (R3) page 15: enhance with images of "small targets" and "low contrast objects" the paragraph that begins: We point out that the evaluators ... (sentences added to address this which refers to a prior figure.)