A grammatical approach to mine detection

Christopher Raphael* Stuart Geman†

March 25, 1997

ABSTRACT

We demonstrate a three stage procedure for mine detection. The first phase locates plausible candidates through a series of hypothesis tests. The second state finds the globally optimal convex set containing each candidate using "coarse-to-fine" dynamic programming. The third stage refines the mine shape estimates using the knowledge contained in a mine-shape library.

Keywords: mine detection, dynamic programming, convex set, global optimization, coarse-to-fine

1 Introduction

In this document we describe our work on mine detection in which, given an optical digital image of a scene, we try to locate and classify the mines that are present. We assume that in our images a mine appears as a region of comparatively brighter pixels (or darker pixels in some imagery), the region belonging to a known library of possible shapes. Nothing is assumed about the possible orientation or location of the shapes and we make only minimal assumptions regarding their size.

Let X be the lattice of pixel locations and $f(x)$, $x \in X$ be the corresponding vector of grey-level intensities. Suppose the possible mine shapes are parametrized by some set Θ, so $\theta \in \Theta$ defines shape type, location, and shape parameters (eg. circle, center, radius) but not parameters relating to grey-level. For $\theta \in \Theta$ we define $I(\theta) \subset X$ to be the interior of the region parametrized by θ and let $O(\theta) \subset X$ be a small "band" around $I(\theta)$. We assume that the grey-level is more or less constant across the $I(\theta)$ and $O(\theta)$ for any mine in the image.

For any $\theta \in \Theta$ we define

$$\begin{aligned} m_I &= \frac{1}{|I|}\sum_{x\in I} f(x) \\ m_O &= \frac{1}{|O|}\sum_{x\in O} f(x) \\ s^2 &= \frac{\sum_{x\in I}(f(x)-m_I)^2 + \sum_{x\in O}(f(x)-m_O)^2}{|I|+|O|-2} \end{aligned}$$

where $I = I(\theta)$ and $O = O(\theta)$. We then define the contrast statistics

$$C(\theta) = \frac{m_I - m_O}{s} \tag{1}$$

$$C(x) = \max_{\theta: x\in I(\theta)} C(\theta) \tag{2}$$

*Department of Mathematics and Statistics, University of Massachusetts, Amherst, email: raphael@math.umass.edu

†Division of Applied Mathematics, Brown University, email: geman@brownvm.brown.edu

It is intuitively clear that, under suitable assumptions, points lying in the interior of mines will be local optima of $C(x)$ and for these interior points, the maximizing θ in the definition of $C(x)$ will be the true mine shape. Given these observations we propose the following idealized recognition strategy. For each $x \in X$ we compute $C(x)$ and if $C(x) > \tau$, for some threshold τ, we say we have recognized a mine and let the maximizing θ be the estimated mine shape.

The practical challenges in performing our procedure are not to be dismissed, however. It is computationally burdensome to compute $C(x)$ at a *single* location and the procedure, as stated, is certainly infeasible. We now describe a three step method designed to approximate this scheme.

In the first stage (Section 2), we visit every site, x, in the image in search of candidates worthy of further consideration. At each site we consider a small number of data points near x. If one can argue that this data is highly unlikely when $C(x) > \tau$, the site is rejected from further consideration. Otherwise we consider a similar test based on more data thus having greater statistical power yet demanding greater computational expense. We continue in this manner until a test is failed or all tests are passed, in which case the candidate is deemed worthy of further consideration.

The second stage (Section 3), which is really the heart of our algorithm, begins with the observation that our possible mine shapes (circle, ellipse, parallelogram etc.) are a subset of the class of convex sets. We then seek to estimate an approximation to the optimal mine boundary by maximizing an approximation to $C(\theta)$ over the class of *convex* sets containing x. Due to the grammatical nature of the class of convex sets, we can compute both the *globally* maximal value of the contrast measure and the associated convex set with a variation on dynamic programming. We use the convex set estimate as input into the third stage.

In the third stage (Section 4), we approximate $C(x)$ itself for the surviving candidates. For each possible shape, we use our convex set estimate to determine starting parameters (e.g. center, orientation, major and minor axes for an ellipse) and proceed to optimize over these parameters using a local search technique. The success of this phase relies heavily on having a good estimate to begin with so the previous phase in indispensable. Our estimate of $C(x)$ is then the maximum value we can achieve by optimizing $C(\theta)$ over θ. If our estimate of $C(x)$ is less than τ we reject the candidate; otherwise we classify it according to the shape associated with the optimizing value of θ.

Finally, Section 5 details our experimental results.

2 Locating Candidates

We demonstrate here how to prune the large number of possible mine locations down to a computationally feasible size.

Let

$$\begin{aligned} D(x,r) &= \{y \in X : |y-x| \le r\} \\ A(x,r_1,r_2) &= \{y \in X : r_1 < |y-x| \le r_2\} \end{aligned}$$

and choose $r_{\max}, R_{\min}, R_{\max}$ so that for every mine hypothesis $\theta \in \Theta$ there is an $x \in X$ satisfying

$$D(x, r_{\max}) \subset I(\theta) \tag{3}$$

$$I(\theta) \cap A(x, R_{\min}, R_{\max}) = \emptyset \tag{4}$$

(the situation of Eqns. 3-4 is depicted in Fig. 1).

Define the contrast measure $C_0(x)$ using $I = D(x, r_{\max})$ and $O = A(x, R_{\min}, R_{\max})$ in Eqn. 1. Note that when the conditions of Fig. 1 are satisfied, $C_0(x)$ is a good approximation to $C(x)$. We will test the hypothesis that $C_0(x) \ge \tau$ at every location in the image. Since the conditions of Fig. 1 are satisfied for at least one x for each

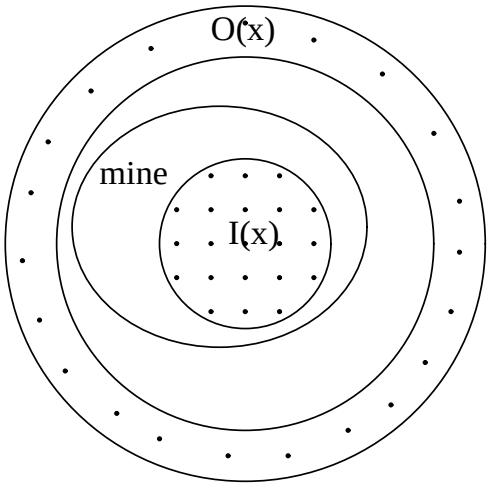

Figure 1: A test consists of choosing a collection of points from the "inside" and "outside" regions. The contrast statistic $\hat{C}_0(x)$ is formed in an attempt to defeat the hypothesis that a mine is present.

mine, and we have assumed that $C(x) > \tau$, we can control the probability of false rejection of a mine. At this stage we concede a certain number of false positives since there will be regions of high contrast but no discernible mine shape and we cannot hope to reject these without an estimate of the region shape.

Let $I(x)$ and $O(x)$ be N-point subsets of $D(x, r_{\max})$ and $A(x, R_{\min}, R_{\max})$ and define $\hat{C}_0(x)$ as in Eqn. 1. Under normal and independence assumptions $\hat{C}_0(x)$ has approximately a noncentral t distribution

$$\hat{C}_0(x) \sim \sqrt{\frac{2}{N}} t(C_0(x)\sqrt{\frac{N}{2}}, 2N-2)$$

where the 1st argument is the noncentrality parameter and the 2nd argument is the degrees of freedom. Thus we can choose $t(\alpha)$ such that

$$P(\hat{C}_0(x) > t(\alpha)) = 1 - \alpha$$

when $C_0(x) > \tau$. Our objective is to reject as much of the image as possible with a minimum of computation effort. To this end we define K (≈ 5) uniformly distributed nonintersecting subsets of $D(x, r_{\max})$ and $A(x, R_{\min}, R_{\max})$, $\{I_k(x), O_k(x)\}_{k=1}^{K}$, so that $|I_k(x)| = |O_k(x)| = N_k$. At each location in the image we perform the aforementioned test sequentially for $k = 1, 2, \ldots, K$ with $\hat{C}_0(x)$ based on the sets $I_k(x)$ and $O_k(x)$. If a point fails one of the tests it is rejected from further consideration.

Typically when one point passes the sequence of tests, so do many of its neighbors ($C(x)$ is constant on the interior of a mine and $C_0(x)$ approximates $C(x)$). To avoid considering many slightly different versions of what is essentially the same location, we cluster the candidates and pass only the cluster centroids to the next stage of the process as a candidate mines.

3 Finding Convex Sets

Upon completion of the first phase of our algorithm, we are left with a list of candidate mine locations. For each candidate we now seek the convex set containing the candidate location and maximizing the contrast measure. In this section we develop a dynamic programming algorithm that, up to a discretization, computes the *globally* optimal convex set containing the candidate. The computational burden of this approach is prohibitive at a reasonable discretization, so we introduce a variation on dynamic programming that, when adapted to our algorithm, allows a computationally feasible means of approximating the optimal convex set.

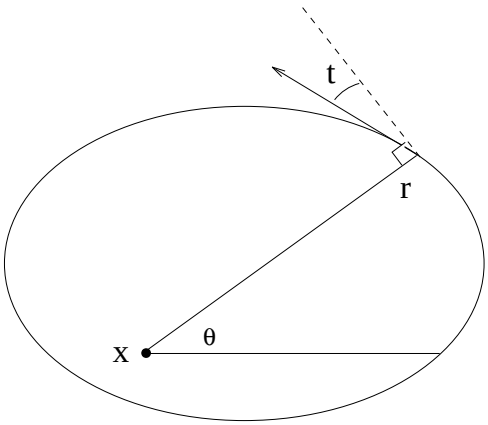

Figure 2: A convex set containing x can be described in terms of the function $r(\theta)$ giving the distance to the boundary as a function of θ. $t(\theta)$ is the relative orientation of the tangent direction to the convex set at the boundary point $(\theta, r(\theta))$.

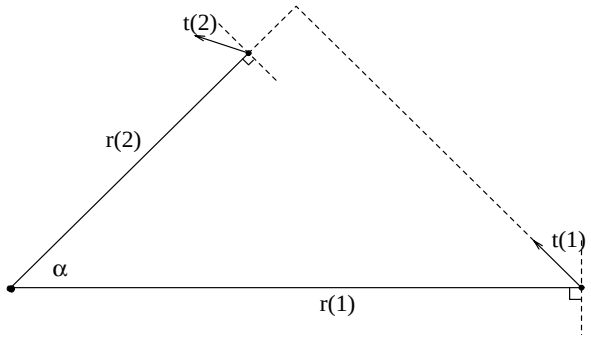

Figure 3: The first of the two conditions for a legal state is described pictorially. For convexity, $t(\theta)$ must rotate counter-clockwise as θ increases. This places a constraint on $r(\theta_1)$, $r(\theta_2)$, $t(\theta_1)$, and $t(\theta_2)$.

3.1 Context-Free Representation of Convex Sets

The family of convex sets containing a fixed reference point is formidably large; before one can hope to find a globally optimal member of this class, one must first represent or describe the class in a meaningful way. We develop a grammatical description of the family of convex sets containing a fixed reference point $x \in \Re^2$. Although this representation seems best suited for *generating* convex sets, it adapts quite naturally to the task of *recognizing* them.

Let $r(\theta) > 0$ for $0 \leq \theta < 2\pi$ be a function describing the boundary of a convex set in polar coordinates regarding x as the origin. Similarly, let $-\pi/2 \leq t(\theta) \leq +\pi/2$ for $0 \leq \theta < 2\pi$ describe the orientation of the tangent to the convex set at $r(\theta)$ measured so that $+\pi/2$ points in toward x and $-\pi/2$ points away from x. (see Fig. 2).

It is clear that $r(\theta)$ and $t(\theta)$ define a convex set containing x if and only if the non-relative tangent angle $t(\theta) + \theta$ is nondecreasing in θ with $r(0) = r(2\pi)$. We now describe a "grammatical" procedure for generating convex sets containing x.

We begin by choosing the values of $r(\theta)$ and $t(\theta)$ at $\theta = 0$ and $\theta = \pi$ — clearly there are no constraints on these choices. We then choose the values of $r(\theta), t(\theta)$ at $\theta = \pi/2$ and $\theta = 3\pi/2$, but now according to certain constraints which will be described presently. The next level chooses $r(\theta), t(\theta)$ for $\theta \in \{\pi/4, 3\pi/4, 5\pi/4, 7\pi/4\}$ and we continue in this way so that after n levels of the derivation the boundary is specified at 2^n locations.

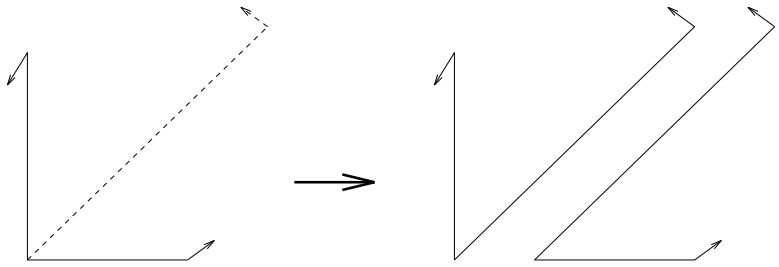

Figure 4: A possible example of the production $s_\sigma \to s_{\sigma\circ 0}, s_{\sigma\circ 1}$. In the children states the two "inner" half-states are shared and the two outer half-states are inherited

If we consider a piece of a convex boundary, say between θ_1 and θ_2, there are several constraints on the values of $r(\theta_1), t(\theta_1), r(\theta_2), t(\theta_2)$. Referring to Figure 3, it is clear that the line emanating from the (polar coordinate) point $(\theta_1, r(\theta_1))$ in the $t(\theta_1)$ direction must intersect the ray emanating from x in the θ_2 direction at a distance less than $r(\theta_2)$ from x. This constraint and the symmetrical constraint generated by reversing the roles of θ_1 and θ_2 lead to the inequality

$$\frac{\cos(t(\theta_1) - (\theta_2 - \theta_1))}{\cos(t(\theta_1))} \leq \frac{r(\theta_1)}{r(\theta_2)} \leq \frac{\cos(t(\theta_2))}{\cos(t(\theta_2) + (\theta_2 - \theta_1))} \tag{5}$$

which must be satisfied by any $r(\theta_1), r(\theta_2), t(\theta_1), t(\theta_2)$ describing a piece of a convex boundary. With this in mind we introduce the notion of a *legal state*, s, as a 4-tuple

$$s = (r(\theta_1), t(\theta_1), r(\theta_2), t(\theta_2)) \tag{6}$$

satisfying Eqn. 5. In the grammatical representation we develop here, a legal state can "divide" into two "child" states by choosing the values of $r(\theta)$ and $t(\theta)$ at some intermediate point θ_{int}, $\theta_1 < \theta_{\text{int}} < \theta_2$. In doing so the "inner" angle radius-tangent pair is shared while the "outer" angle radius-tangent pairs are inherited, that is,

$$(r(\theta_1), t(\theta_1), r(\theta_2), t(\theta_2)) \longrightarrow (r(\theta_1), t(\theta_1), r(\theta_{\text{int}}), t(\theta_{\text{int}})), (r(\theta_{\text{int}}), t(\theta_{\text{int}}), r(\theta_2), t(\theta_2))$$

(see Fig.4). If these productions are to result in a convex set it is clear that the child states must also be legal states. In fact, this condition is necessary *and* sufficient. We will call such a production a *legal production.*

We define our grammatical derivation on the diadic points of $[0, 2\pi)$. To this end, we define the index set $\{0,1\}^*$ to be the finite length strings on $\{0,1\}$ and for each $\sigma \in \{0,1\}^*$ we define the state s_σ as the one corresponding to the interval

$$(\theta_1, \theta_2) = (\frac{2\pi b(\sigma)}{2^{|\sigma|}}, \frac{2\pi(b(\sigma)+1)}{2^{|\sigma|}})$$

where $|\sigma|$ is the length of σ and $b(\sigma)$ is the number in $\{0, 1, \ldots, 2^n - 1\}$ obtained by viewing σ as a binary number. Our derivation then begins by choosing $\sigma_\emptyset = (r(0), t(0), r(2\pi), t(2\pi))$ with $r(0) = r(2\pi)$ and $t(0) = t(2\pi)$ and then continuing recursively with legal productions of the form

$$s_\sigma \to s_{\sigma\circ 0}, s_{\sigma\circ 1} \tag{7}$$

for $\sigma \in \{0,1\}^*$ (see Fig. 6).

If one were to terminate the derivation at level n and consider the possible collection of 2^n states $\{s_\sigma : |\sigma| = n\}$ produced by the grammar, one could associate a convex polygon with this partial derivation by connecting the endpoints of the $\{(\theta_k, r(\theta_k)) : k = 0, 1, \ldots, 2^n - 1\}$ specified by the $\{s_\sigma\}$ (see Fig. 5). (It is simple to show that this set is in fact convex.) By letting the derivation continue indefinitely and taking the limit of the increasing sequence of convex polygons as the result of the derivation, we have constructed a grammar for the entire family of

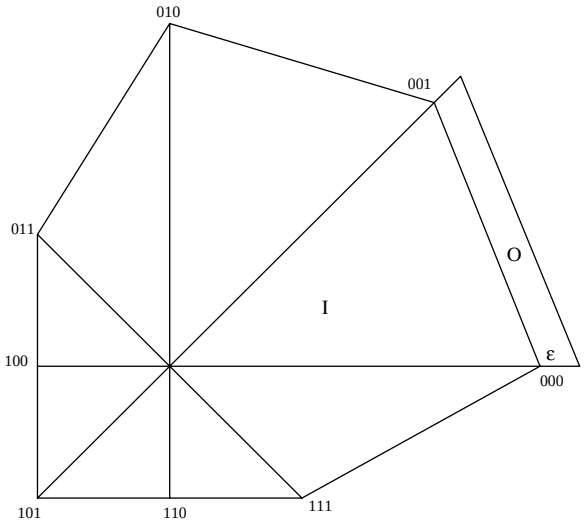

Figure 5: The inside ($I(s_\tau)$) and outside ($O(s_\tau)$) regions used in computing the contrast measure for a terminal state.

convex sets containing x. That the result of our derivation is a convex is clear since it is the limit of an increasing sequence of convex sets. To see that *any* convex set can be obtained in this way, one need only observe that a fixed convex set will "force" a sequence of productions which are all legal.

In practice, we seek realizable computer algorithms, so we do not carry out the construction indefinitely. Typically we stop after 5 or 6 levels, thus specifying 32 or 64 boundary locations and tangent angles for the convex set. In the sequel we will denote this *terminal level* by N and call the *state index set*

$$\Sigma = \{\sigma \in \{0,1\}^* : |\sigma| \leq N\}$$

and the *terminal state index set*

$$\Sigma_T = \{\tau \in \Sigma : |\sigma| = N\}$$

A *terminal state* is one indexed by $\tau \in \Sigma_T$. We will write $s = \{s_\tau : \tau \in \Sigma_T\}$ for a convex set generated by our grammar. In Fig. 6, the terminal states, (the lowest row of tree vertices) show an example of such a convex set.

3.2 The Recognition Algorithm

Suppose $x \in X$ is a reference point allegedly contained in some convex set depicted in our image f. We now look at the problem of *identifying* the "best" convex set containing x.

For each terminal state ("pie slice"), $s_\tau \in \Sigma_T$, let $I(s_\tau)$ and $O(s_\tau)$ be the regions depicted in Fig. 5. Define the contrast measure $C(s_\tau)$ in the usual way following Eqn. 1. For a convex polygon $s = \{s_\tau : \tau \in \Sigma_T\}$, (a possible derivation of our grammar) we define the total contrast measure to be

$$C(s) = \sum_{\tau \in \Sigma_T} C(s_\tau)$$

This definition is not the same as the "true" contrast we would achieve by using Eqn. 1 on the interior and exterior of the entire convex set since we allow the m_I,m_O, and s^2 to vary from state to state. It should rather be taken as a more easily optimizable approximation to the true contrast.

We now demonstrate an algorithm that produces a *global* maximum of the contrast measure over the subset of convex sets created by stopping our grammatical derivation after N levels.

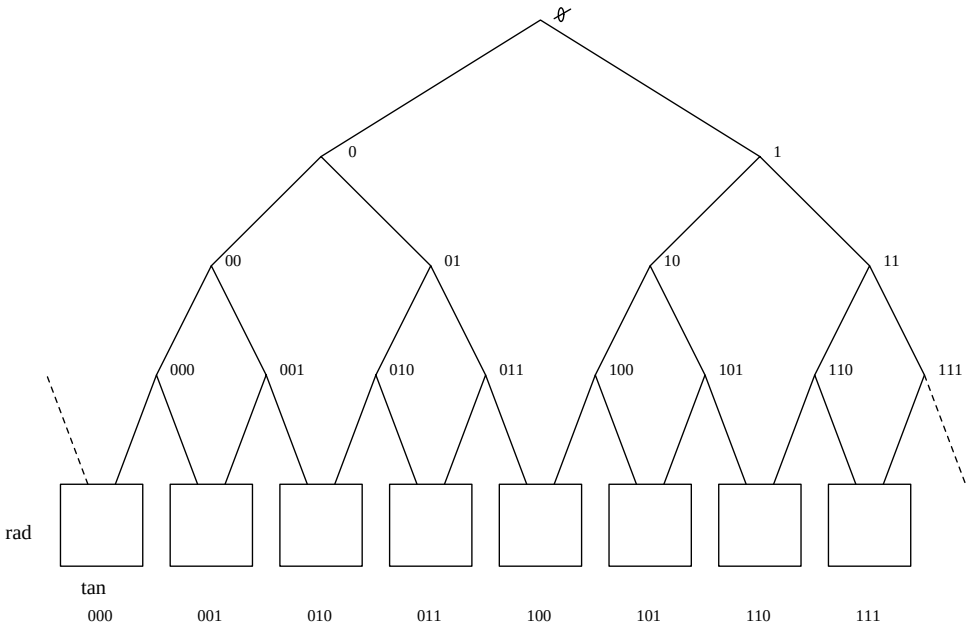

Figure 6: The tree-like derivation of a convex set. Each tree node is labeled with its index. The bottom row of boxes denote the radius-tangent spaces. Each box corresponds to an angle, θ, at which the radius and tangent are specified. In the figure that is $\theta = 2\pi k/8$ for $k = 0, 1, \ldots, 7$ (from left to right). In the grammatical derivation, the choice of state at level 0, $(s_\emptyset)$, fixes the radius and tangent at $\theta = 0$ (the point in the $k = 0$ box). The production $s_\emptyset \rightarrow s_0, s_1$ specifies the radius and tangent at $\theta = \pi$ (the point in the $k = 4$ box) etc.

For each legal terminal state, s_τ, we begin by computing $C(s_\tau)$ for each legal value of s_τ. For each s_σ at the penultimate level ($|\sigma| = N - 1$) there are a number of possible legal productions following from s_σ. Each one of these production could be scored by summing the contrast scores of the left and right children. We define the optimal contrast for s_σ, $C(s_\sigma)$, to be the maximum contrast sum over all possible productions:

$$C(s_\sigma) = \max_{s_\sigma \rightarrow s_{\sigma\circ 0}, s_{\sigma\circ 1}} C(s_{\sigma\circ 0}) + C(s_{\sigma\circ 1}) \tag{8}$$

and we will "remember" the optimizing production associated with each s_σ. It is clear that, if s_σ occurs in our derivation, then $C(s_\sigma)$ is the maximum contribution of children of s_σ to the contrast measure. This reasoning applies equally well to states at all levels of our grammar. If for *all* nonterminal states s_σ we define $C(s_\sigma)$ recursively by Eqn. 8, then we can compute our optimal contrast measure at progressively earlier states (higher levels in Fig. 6) of the grammar. Thus for *any* s_σ, $C(s_\sigma)$ is the maximum contribution of all progeny of s_σ to the contrast measure. Ultimately we will be able to compute $C(s_\emptyset)$, the maximum constrast measure over all legal derivations following from $s_\emptyset$. The optimal contrast over all derivable convex sets is then

$$C^* = \max_{s_\emptyset} C(s_\emptyset) \tag{9}$$

Having found the optimal contrast, it is a simple matter to recover the optimal convex set. The optimal root state is given by

$$s_\emptyset^* = \arg\max_{s_\emptyset} C(s_\emptyset)$$

and, since for every state we store the optimal production rule following from that state, we need only trace these optimal productions from $s_\emptyset^*$ down to the terminal level. The sequence of optimal terminal states then gives the optimal convex set.

As described, the algorithm cannot be performed on a computer since the maximization of Eqn. 9 involves an infinite number of possible productions. We could, however, discretize the state spaces and perform the algorithm exactly as stated. Although simple enough in principle, the computational burden of this discrete approach would be substantial, even for relatively coarse discretization. If we let A be the number of terminal angles, ($A = 2^N$), and H the number of discretized radius-tangent pairs at each angle, then the algorithm is easily seen to be $O(AH^3)$. Allowing merely 10 different radii and tangents for each angle already leads to tens of millions of operations. Some more efficient way of carrying out, or approximating, this computation is necessary if we want our approach to be computationally feasible.

3.3 Coarse-to-Fine Dynamic Programming

Consider the familiar dynamic programming scenario depicted in the upper left of Fig. 7. Associated with each arc in the graph is a cost, and we seek the minimal cost path from the start state to the finish. The familiar dynamic programming algorithm solves this problem by recognizing that the optimal path from the start state to any interior state at level l must be an extension of some optimal path from the start state to a state at level $l-1$ by an arc connecting the state at level $l-1$ to a state at level l. Imagine a situation where there are so many different states that it is infeasible to perform full-fledged dynamic programming. We propose a modification that in many cases produces the same optimal path in considerably less time.

We build from our original graph, a coarse graph by grouping states together into what we will call "super-states". For the sake of argument, we begin by dividing the states at each level into two super-states. In our coarse graph, any two super-states are connected if any pair in their cross product has an arc (upper right of Fig. 7). Similary, the cost of an arc between two super-states is the *minimal* cost of the arcs in the cross product. We then can find the minimum cost path through the coarse graph with the usual dynamic programming technique with very little computational effort.

Along this optimal path we now refine the coarse graph, say, by splitting each super-state into two smaller super-states. With this new collection of super-states we rebuild the connectivity and arc scores by the same rules as before and perform dynamic programming on this new less coarse graph. In general, this new optimal path (middle left of Fig. 7) will not be a refined version of the previous optimal path and might visit some of the coarser super-states. This process continues until we find an optimal path that lies entirely at the original discretization (lower right of Fig. 7). When this has happened, we have found the optimal path through the original graph. Our optimal path by definition beats any other path in the final coarse graph. However, since the path scores in the coarse graph are better than their original graph counterparts, we have found the optimal path through the original graph as well.

There is no guarantee that this algorithm takes less time to compute the optimal path than does the regular dynamic programming algorithm. For the algorithm to function well, the refining process must involve a minimum of iterations. This requires that the super-states consist of collections of "similar" states, so that the optimistic scores we use for arcs in the coarse graph are not far from the true scores of the constituent arcs.

In many applications the significant computational cost involved in computing the minimum cost between two super-states will defeat the purpose of coarse-to-fine dynamic programming. We remark that our algorithm will produce the optimal path through the graph when we replace this minimum by a *lower bound* for the minimum as long as the lower bound collapses to the true cost for singleton super-states. Choosing this lower bound is a matter of great practical importance. An excessively "loose" lower bound can cause the algorithm to explore much irrelevant territory through unwarranted optimism in the scoring of a coarse path.

With a simple modification this algorithm can be extended to cover cases in which the state space is *continuous*. In this situation, our super-states are given by *intervals* rather than finite sets and the lower bounds which represent arc scores must be taken over the cross product of two such intervals. Each iteration of the algorithm now involves an application of dynamic programming followed by a refinement of the collection of optimal intervals

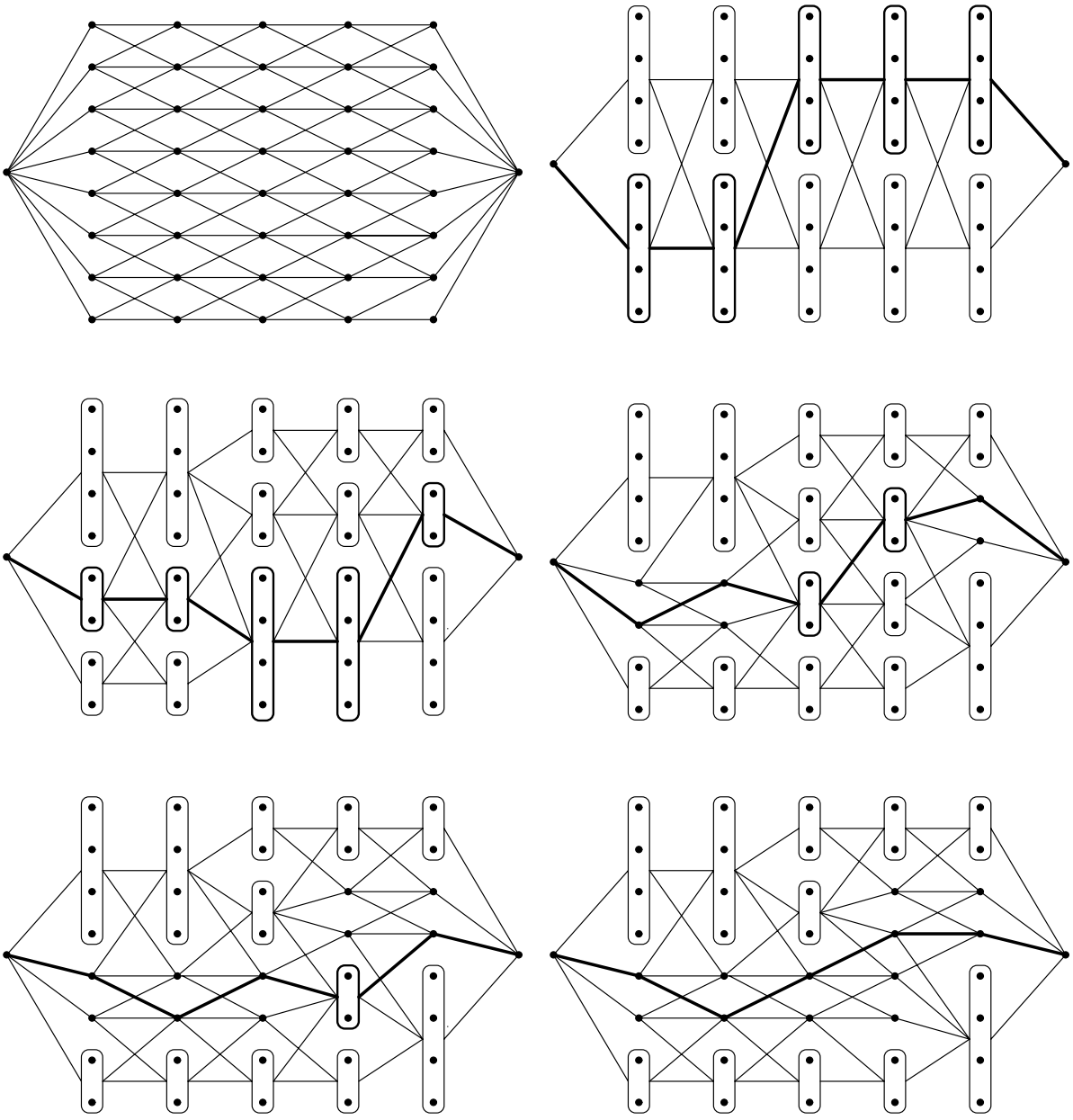

Figure 7: The upper left figure is the original dynamic programming setup. The 5 remaining figures show a possible progression of our coarse-to-fine dynamic programming algorithm. The optimal path (shown in bold lines) in the final figure in the lower right must also be the optimal path in the original lattice.

(state regions) into collections of sub-intervals and a recomputation of the arc scores. Thus each iteration of he algorithm generates a sequence of state intervals. It can be shown, under suitable assumptions, that the state intervals converge to the optimal states as the algorithm is iterated.

3.4 The Recognition Algorithm Revisited

We now combine this *continuous* coarse-to-fine dynamic programming with our convex set recognition algorithm. We begin by partitioning the radius-tangent spaces associated with each terminal angle of our convex set. Let $r_{\min}$ and $r_{\max}$ be the smallest and largest radii we are willing to consider and $t_{\min} = -\pi/2, t_{\min} = \pi/2$. We partition the set $[r_{\min}, r_{\max}) \times [t_{\min}, t_{\max})$ into a collection of similar rectangles as in the top of Fig. 8. At each vertex of the tree we form the super-states which are pairs of radius-tangent rectangles. Each super-state contains a left rectangle and a right rectangle taken from the corresponding left and right radius-tangent spaces. Following the discussion of Section 3.3, we would like to score each super-state in the most optimistic way. That is, if $\mathbf{s}_\tau$ denotes a super-state,

$$C(\mathbf{s}_\tau) = \max_{\substack{s_\tau \in \mathbf{s}_\tau \\ s_\tau \text{ legal}}} C(s_\tau). \tag{10}$$

Using this contrast measure on super-states, we can compute the (optimistic) optimal contrast, C^*, by following the recursion of Eqn. 8 and using Eqn. 9. This optimal contrast is achieved by a sequence of radius-tangent rectangles as depicted in Fig. 8. We now refine only these rectangles, shown in the bottom of Fig. 8. We again form our super-states as before on the new partition of the radius-tangent spaces and score the new super-states through Eqn. 10. Again we find the (slightly less optimistic) optimal contrast using the dynamic programming recursion and the associated optimal sequence of radius-tangent rectangles is depicted in the bottom of Fig. 8. This process continues until all of the radius-tangent rectangles along the optimal path have been refined to a sufficiently small size. At this point we take this sequence as an approximation to our optimal convex set.

The actual computation of Eqn. 10 is formidable and the computational burden of finding this maximum would defeat the purpose of coarse-to-fine dynamic programming. As remarked in the previous section, we can substitute an upper bound for the maximum of Eqn. 10 as long as this upper bound approaches the true maximum as the super-states become small. We have used the upper bound

$$C^{\text{upper}}(\mathbf{s}_\tau) = 1_{\{\mathbf{s}_\tau \text{ contains a legal state}\}} \max_{s_\tau \in \mathbf{s}_\tau} C(s_\tau). \tag{11}$$

The characteristic function is easy to compute using Eqn. 5 and we can easily compute the maximum since it depends only on the two radii ranges. We substitute Eqn. 11 for Eqn. 10 in the actual dynamic programming calculations with the observation that they differ by less and less as the optimal radius-tangent rectangles shrink.

4 Finding Optimal Mine Hypotheses

Once we have a good estimate of the boundary of a convex set, it is a simple matter to refine this estimate using the knowledge contained in our shape library. For each possible shape type (circle, ellipse, parallelogram, etc.) we form an initial estimate of the parameters, θ, of the shape using the boundary estimate. For instance, for the circle we would estimate a center and radius. We then define $I(\theta)$ to be the interior of the shape defined by θ and $O(\theta)$ to be a narrow "band" around the shape and define the contrast measure $C(\theta)$ in the usual way following Eqn. 1. We then use a local optimization technique to find the best parameters for each member of the shape library subject to some simple constraints prohibiting degeneracy of the shape. The shape type and parameter estimate maximizing the contrast measure is taken to be our optimal hypothesis, and the optimal contrast measure is our estimate of $C(x)$. We reject the "mine present" hypothesis if our estimate of $C(x)$ fails to exceed our threshold τ. Otherwise we classify according to the optimal shape type.

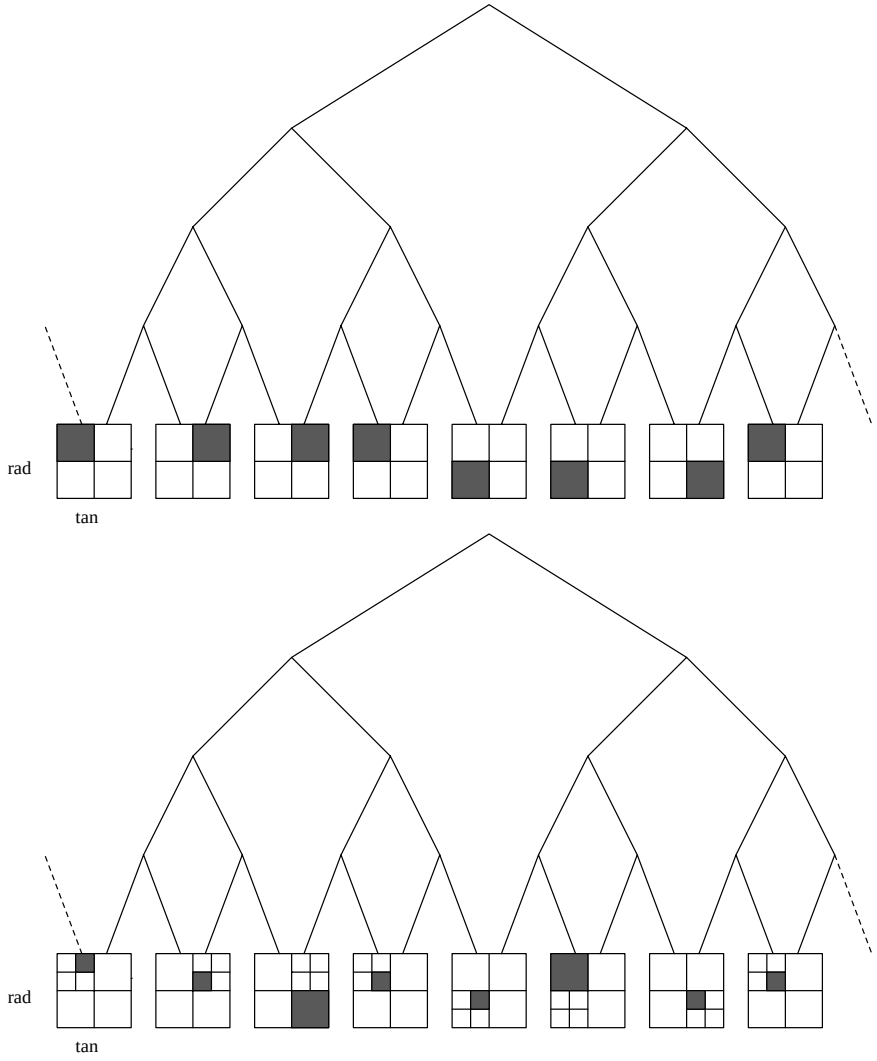

Figure 8: Coarse-to-Fine dynamic programming combined with convex set recognition. In the top figure the sequence of radius-tangent rectangles having maximal optimistic contrast is shown in dark. In the bottom figure we have refined the radius-tangent spaces by partitioning the optimal radius-tangent rectangles into four children. The bottom figure shows the optimal radius-tangent rectangles at this new discretization. This process is continued until the optimal radius-tangent rectangles have reached uniformly a sufficiently fine level of discretization.

5 Experimental Results

We present the results of our algorithm applied to images taken from three different data sets demonstrating a variety of scenarios. The mine objects, although not actual mines, encountered in these images are representative of the size, shape and reflectivity one would encounter with genuine mines.

Our first example describes the performance of our algorithm on a set of 10 256x256 images depicting mines in dunes taken from the "amdas" data set. We knew *a priori* that the images contained only circular mines so our shape library is limited to a single example here. We chose values $r_{\max} = 8$, $R_{\min} = 20$, and $R_{\max} = 25$ (Section 2), thereby presuming that mines we encounter have radii between 8 and 20 pixels. These values allow some latitude in the placement of our mask of Fig. 1 leading to candidate detections. At each location in the image we then performed a sequence of 5 tests comparing regions of size: 2,4,8,16,32 pixels. These tests attempted to defeat the hypothesis that the contrast was greater than or equal to 2.25 using the methods of Section 2. Although this "candidate threshold" is in principle the same as the "final threshold" (τ), the greater flexibility achieved by the independent thresholds leads to slightly better experimental results. Each of these tests was performed at a significance level of .05. An illustrative image and the results of the sequence of tests applied to the image are shown in Fig. 9. The test-response image (right) is formed of six possible grey levels representing the number of tests a point passed $0, 1, \ldots, 5$ (brighter means more tests passed). The set of image locations passing all 5 tests is then clustered following Section 2 and the centroids of these clusters are shown in the upper-left panel of Fig. 10. In producing these clusters we rule that two points are in the same cluster if their "Manhattan" distance is less than 10 pixels.

We next estimate the optimal convex set containing each candidate using the method of Section 3. We represented a convex set by choosing 32 possible (radius,tangent) pairs ($|\Sigma_T| = 32$). The ultimate discretization we seek is one in which a radius-tangent rectangle is $\pi/16$ radians by about 1.5 pixels. Fig. 10 (upper-right) shows the optimal convex set estimated for each of our candidates. Note that some of these estimates are only "almost" convex due to our stopping the algorithm once a fixed degree of precision is obtained and the approximation to our upper bound function of Eqn. 11. Although it is reasonable to reject some candidates at this point based on our approximation to the optimal contrast, we have found that, having already invested significant effort in a particular candidate, it is worthwhile to complete the shape estimation phase before any further rejection is performed.

We then used these convex sets to estimate centers and radii for the one shape under consideration: the circle. We optimize the contrast measure over the (center, radius) parameters comparing the interior of the circle with the concentric bounding annulus having about 3 times the area as the circle. The optimal circles are then depicted in Fig. 10 (lower-left) with the bounding annuli used in computing the contrast measure. In our experience, the contrast measure, when viewed as a function of the shape parameters, is fraught with many local maxima. Thus a good choice of initial parameters is indispensable to this final optimization stage. We have encountered only a few examples in which the final phase has recovered from a poor initial convex set estimate further reinforcing our belief in the necessity of global optimization in scene-analysis problems. Finally, Fig. 10 (lower-right) shows only the mine hypotheses exceeding our final threshold 3.5.

We repeated this experiment for a second set of 10 images containing mines in shallow water and a representative example is shown in Fig. 11. These images were considerably more difficult than the previous scenario due to the decreased contrast and reflection from the water. The parameters were exactly as before except that the candidate and final thresholds were 2.0 and 1.9.

A final set of 2 720x480 images deals with a more complicated scenario in which a variety of different shapes are possible falling under the general categories of ellipse and parallelogram. Unlike the previous two examples, in these images a mine appears as light pixels on a dark background. The greater range of shape sizes demands a less opinionated choice of parameters in our candidate detection phase where we have chosen $r_{\max} = 8$, $R_{\min} = 40$, and $R_{\max} = 45$. The candidate threshold was 2.25. Evidently, this and the uneven texture of the background

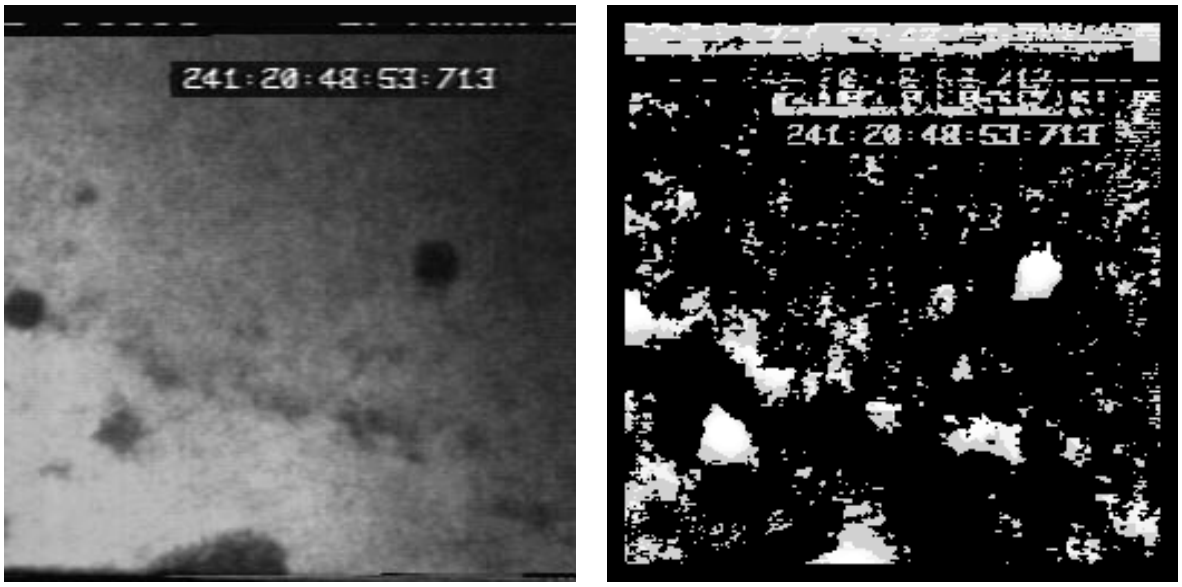

Figure 9: Example 1. *Left:* The original image *Right:* The darkness at a particular location indicates how many of the tests were passed at that location. The brightest grey-level indicates all tests were passed and the darkest grey-level means no tests were passed.

lead to many more candidate locations as depicted in Fig. 12 (top). The optimal convex sets containing these candidates are displayed in Fig. 12 (bottom). Due to the larger area being considered in these estimates, our final resolution is less refined than in the previous examples — the radii are estimated to ranges of about 3 pixels. In performing the final shape estimates, we now optimize over both the shape types: ellipse and parallelogram. For added security we search for parallelograms starting from two different initial parameter choices corresponding to a rotation by 90 degrees. The final shape estimates and the shape hypotheses exceeding the minimal contrast of 2.75 are shown in Fig. 13

6 Acknowledgements

The research reported herin was supported by Contract No. N00014-93-C-0078 from the Office of Naval Research to Mathematical Technologies Inc.

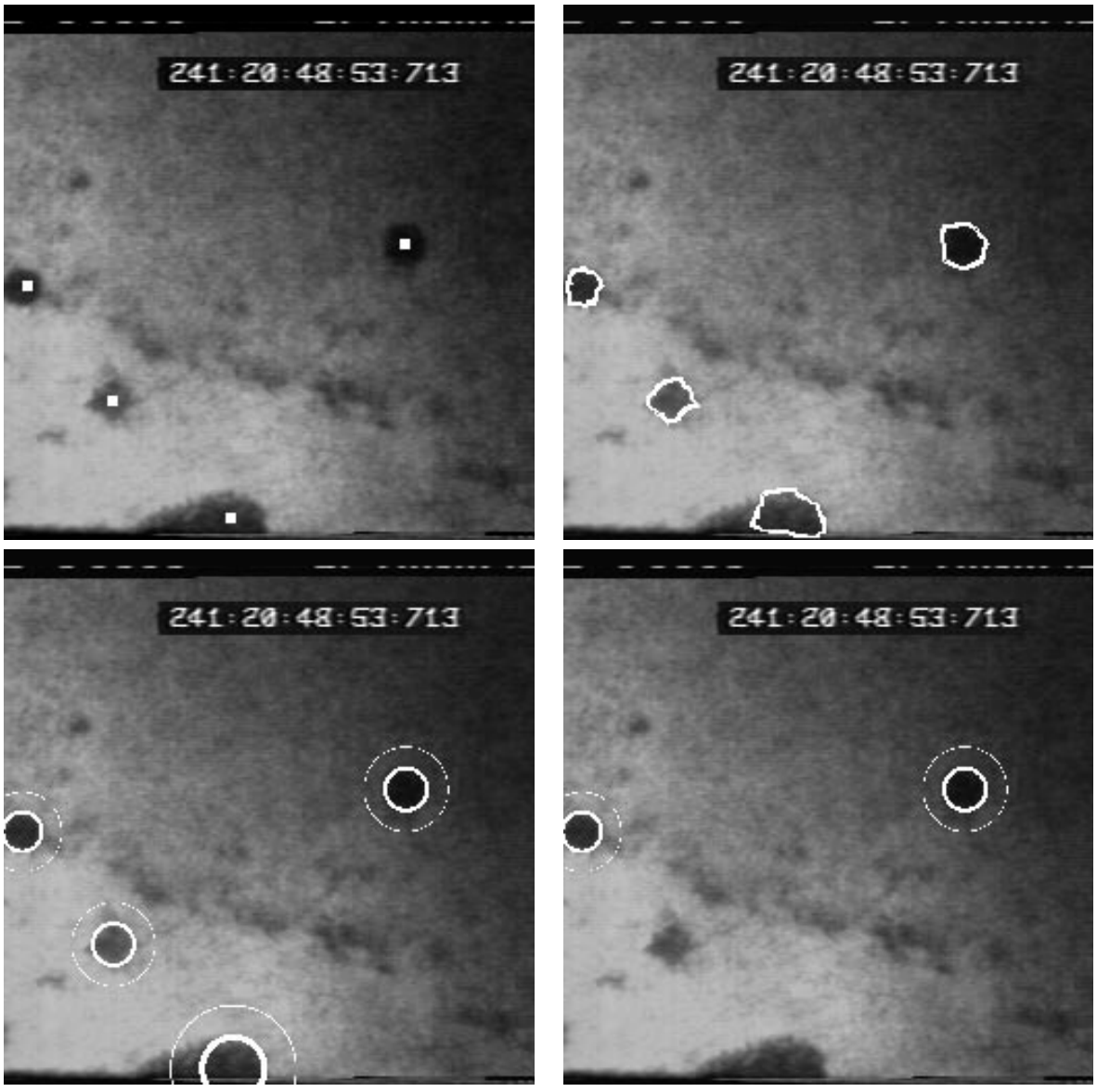

Figure 10: Example 2. *Top Left:* The candidate locations *Top Right:* Optimal convex sets *Bottom Left:* Optimal circles *Bottom Right:* Best surviving hypotheses after thresholding

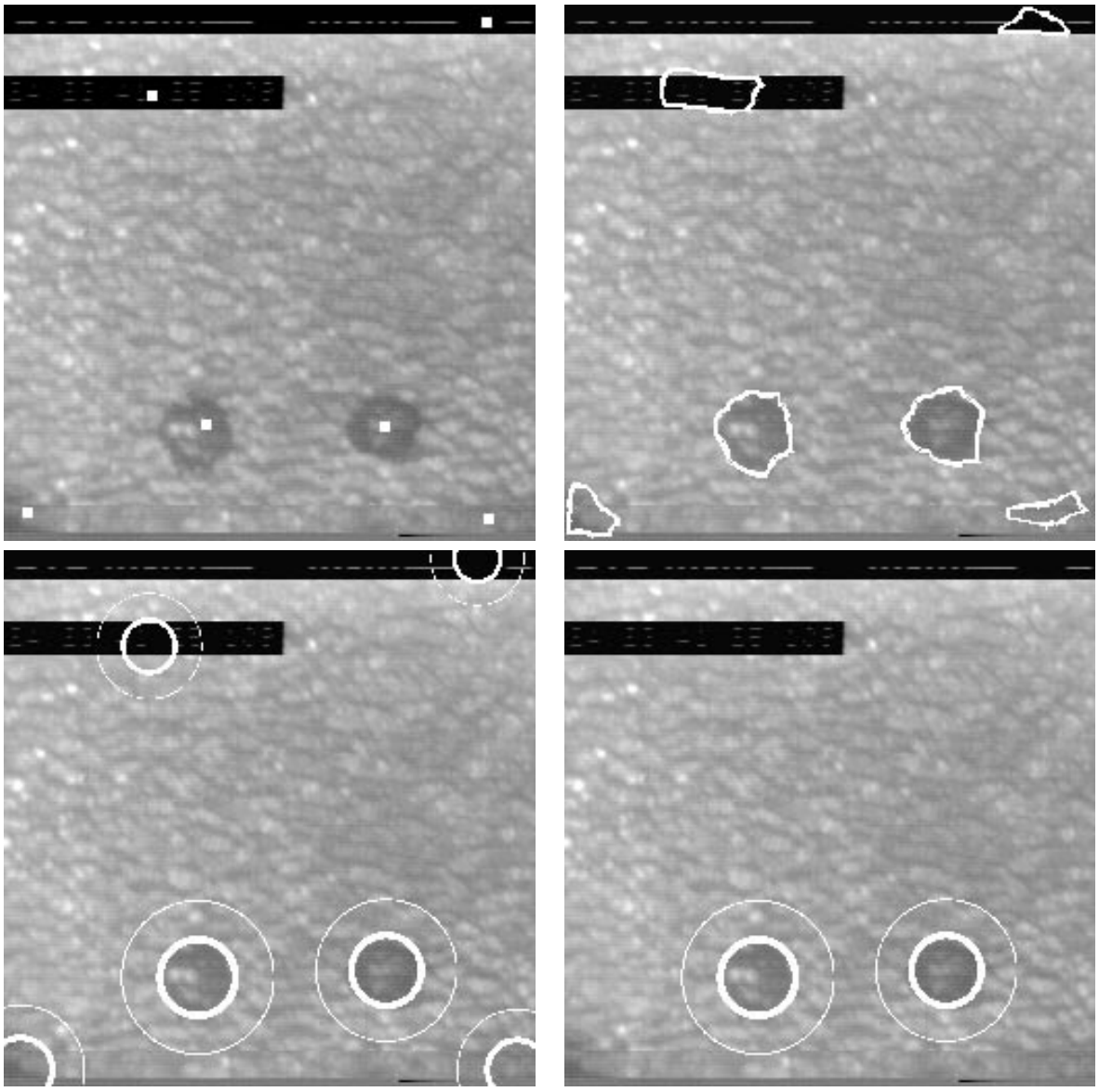

Figure 11: Example 2. *Top Left:* The candidate locations *Top Right:* Optimal convex sets *Bottom Left:* Optimal circles *Bottom Right:* Best surviving hypotheses after thresholding

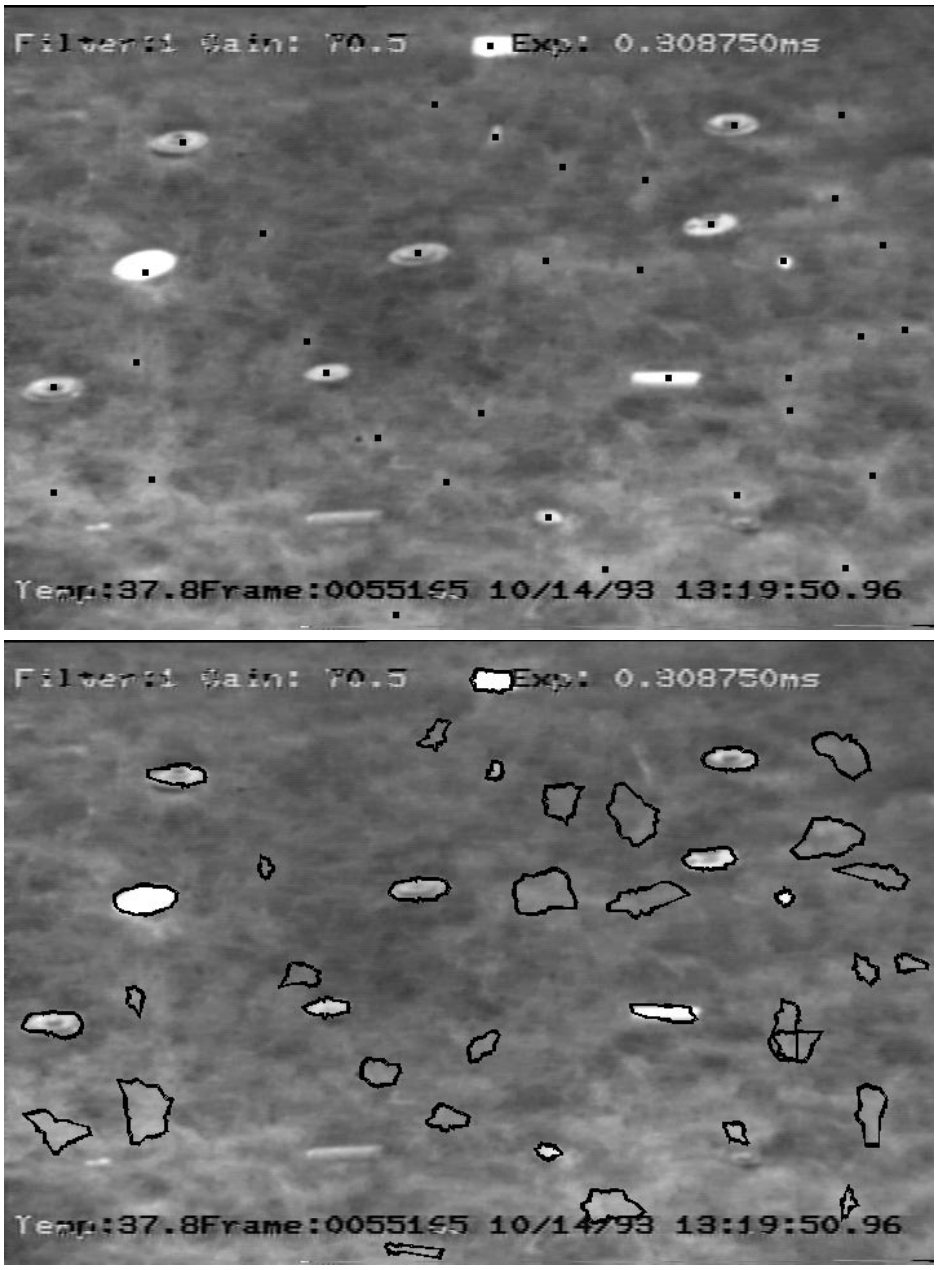

Figure 12: Example 3. *Top:* The candidate locations *Bottom:* Optimal convex sets

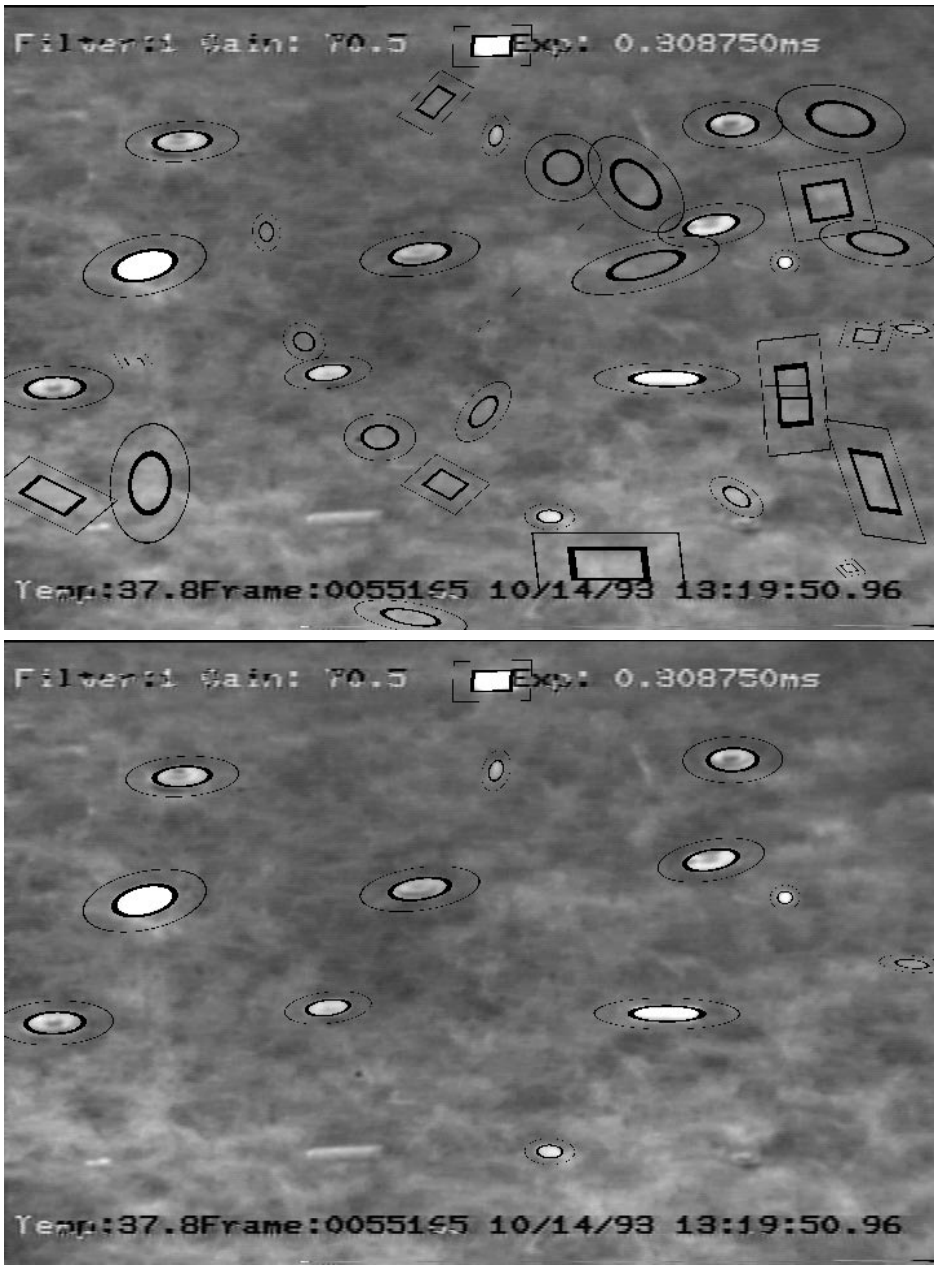

Figure 13: Example 3. *Top:* The optimal shapes *Bottom:* Surviving hypotheses after thresholding.