Efficient Determination of Shape from Multiple Images Containing Partial Information

Ronen Basri[*]
Department of Applied Math.
The Weizmann Institute of Science
Rehovot, 76100, Israel
ronen@wisdom.weizmann.ac.il

Adam Grove and David Jacobs
NEC Research Institute
4 Independence Way
Princeton, NJ 08540, USA
(grove/dwj)@research.nj.nec.com

Abstract

We consider the problem of reconstructing the shape of a 2-D object from multiple partial images related by scaled translations, in the presence of occlusion. Lindenbaum and Bruckstein have considered this problem in the specific case of a translating object seen by small sensors, for application to the understanding of insect vision. Their solution is limited by the fact that its run time is exponential in the number of images and sensors. We generalize the problem to allow for arbitrary types of occlusion of objects that translate and change scale. We show that this more general version of the problem can be solved in time that is polynomial in the number of sensors, but that even the original problem posed by Lindenbaum and Bruckstein is in fact NP-hard when the number of images is unbounded. Finally, we consider the case where the object is known to be convex. We show that Lindenbaum and Bruckstein's version of the problem is then efficiently solvable even when many images are used, as is the general problem in certain more restricted cases.

[*]The research of Ronen Basri was supported in part by the Israeli Ministry of Science, Grant No. 6281. The vision group at the Weizmann Institute is supported in part by the Israeli Ministry of Science, Grant No. 8504. Ronen Basri is an incumbent of Arye Dissentshik Career Development Chair at the Weizmann Institute.

1 Introduction

We consider the problem of reconstructing the shape of a 2-D object using multiple partial images. We assume that some portions of each image can be identified as definitely belong to the object, other portions as definitely belonging to the background, and the remainder which are ambiguous due to occlusions or other uncertainty. In order to solve this problem we must account for the fact that unknown transformations relate the images to each other, because the object or camera undergoes unknown motions between frames. We also do not assume it is possible to identify any specific local features that could be matched between frames and thus used to recover the unknown motion. Therefore we must make use of a relatively weak constraint, namely that we can rule out motions which would assign the same scene position to be figure in one frame and background in another. We show how to determine the set of possible objects and motions consistent with this constraint.

One main motivation of our work has been to generalize and improve upon the results of Lindenbaum and Bruckstein[12](L&B), who have previously raised the problem of determining the shape of a 2-D object from sparse measurements. They considered the case of an unknown object that translates over a set of small visual sensors. In their problem, every sensor reports whenever the boundary of the object passes within the sensor's receptive field, providing a thin slice of the unknown shape. L&B sought an algorithm to combine measurements obtained from the set of sensors during different translations of the same object in order to recover its full 2-D shape. Their primary motivations for considering this problem were to model insect vision systems and to solve industrial problems.

We view this situation as a special case of the problem of shape recovery in the presence of occlusion. The problem defined by L&B corresponds to a situation in which an object is viewed through a grating of narrow slits in a sequence of images related by translation. In this case slices of the shape of the object at the positions of the slits are seen perfectly, but between the slits the object is occluded. Moreover, the slits are too narrow to allow for extracting any features that remain invariant under different translations of the object, since any local feature identified in one view of the object is likely to be occluded in other views. Consequently, combining information from measurements obtained under different translations of the same object cannot rely on matching localized features. In this, L&B's problem is related to an approach to recognition in which the relative pose of a model and image are determined, given a correspondence between regions of the model and image but with no specific correspondence between local geometric features ([4, 11]). We generalize their problem by allowing for more arbitrary types of occlusions. In our formulation, the scene need not be viewed through 1-D slits; any portions of the object and background may be visible. And we will allow for scale changes, in addition to translations, that may relate different images. We will also indicate how one of our algorithms may be extended to a wider range of possible image transformations. Figure 1 illustrates the input to our and L&B's algorithms.

In our more general setting, the problem posed by Lindenbaum and Bruckstein is also closely related to that of tracking and building up a model of an unknown object in the presence of occlusion. Imagine, for example, that one has several pictures of a bird in flight, as seen through the branches of a tree, against the background of the sky. Portions of each image are known to come from the bird. Where the sky appears, we know that the bird's shape is not present. But where one sees branches, there may be either bird or sky behind. By allowing for general patterns of figure/background/occlusion we will account for this type of viewing situation.

In addition to generalizing L&B's problem, we also provide insight into the computational complexity of the problem. L&B developed an algorithm that requires computation that is exponential in both the number of images and the complexity of the description of each image. We provide an algorithm for our more general problem that requires computation that grows as a polynomial function of the image complexity (but still exponential in the number of images); this algorithm is practical when the number of images used is small. We then show that it is apparently not possible to produce a general algorithm that is polynomial in the number of images. Specifically, we show that even L&B's more constrained version of the problem is NP-hard. We then provide an algorithm that *is* polynomial in the number of images for a special case of the problem. The main restriction is that the sensed object must be known to be convex. This algorithm finds a shape consistent with the images by running a linear program. The number of variables in the program is linear in the number of images. The number of linear constraints derived is either linear or quadratic in the complexity and number of images, for two different variations of the problem. In general, the algorithm is practical in many situations of interest.

The paper is divided as follows. In Section 2 we formulate our problem and discuss past work. Then, in Section 3 we introduce a solution with complexity that is polynomial in the size of the individual image representations, but remains exponential in the number of images. In Section 4 we show that the problem is NP-hard in the number of images. Next, in Section 5 we introduce an efficient solution for convex shapes. Finally, we present the results of experiments in Section 6.

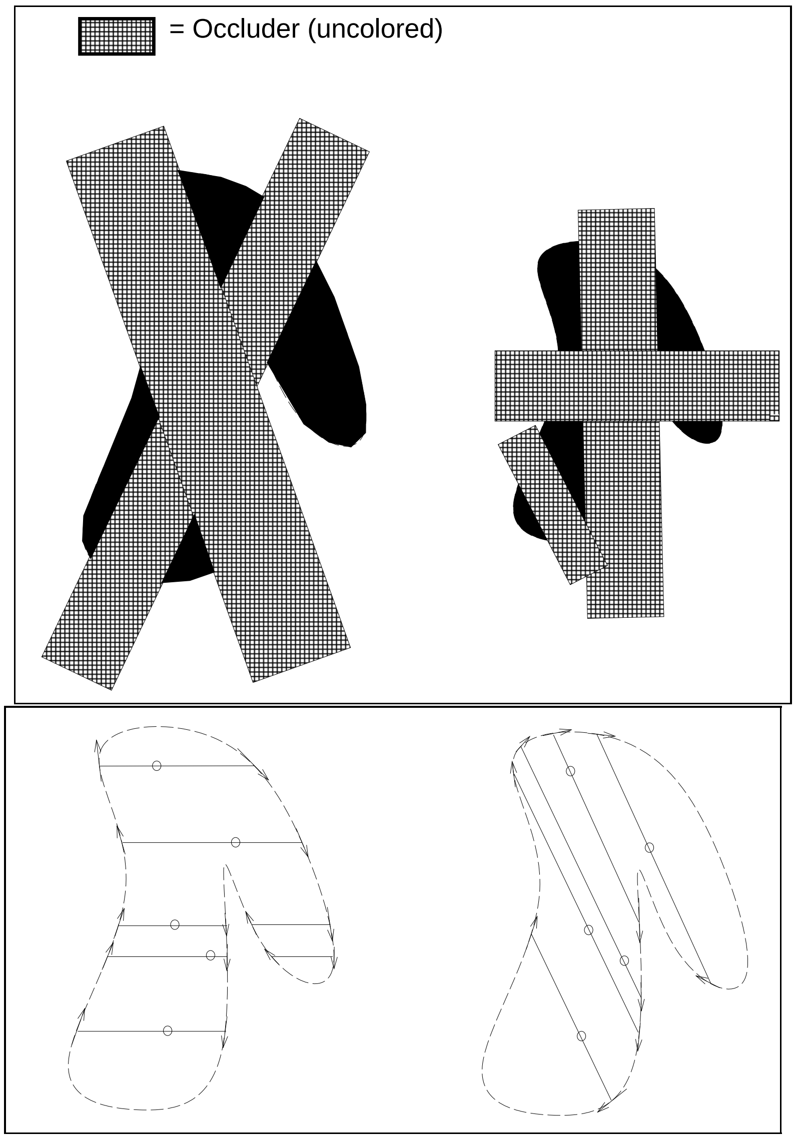

Figure 1: Our algorithms assume that a series of images like the ones on the top must be related by scaled translations. Here, arbitrary subsets of the figure are seen, some of the background is known, and occluders hide portions of the scene, which may be figure or background. L&B's algorithm assumes that the images have the form shown on the bottom, in which the scene is only sensed along lines, where line segments of the figure (along with the tangent direction at boundaries) are known. No scaling is allowed in their formulation.

2 Problem formulation and background

We now formulate our problem more precisely, and relate it to past work. First, we assume that we are given m images, each of which is divided into *figure*, *background*, and *occlusions* that may be either figure or background. We think of the region due to the figure as black, the background as white, and the boundary between black and white regions as grey. Occluded regions are uncolored. In order to make this problem discretely representable, we assume that all black and white image regions are given as polygons, with a total of at most n sides and vertices. Each vertex and side may be black, white, or grey. Note, for example, that it is perfectly possible for all sides and vertices to be grey, as when the entire image is divided into black and white, with no occluded regions.

Next, we assume that the positions of the object in the images are related by translations and changes in scale, while the pattern of occlusions may be arbitrarily different between images. Our goal is to bound the set of feasible transformations relating the images, and the corresponding set of possible object shapes. A transformation will be feasible whenever it places two images in a common reference frame in which white and black regions do not intersect. This will ensure that the same image point is not interpreted as both figure and background. To rule out degenerate transformations in which no portion of the object is seen more than once, we may also bound the magnitude of the allowed transformation, or equivalently, assume that figure may only lie in or within some neighborhood of the image by assuming that it is centered inside a frame of white. Our proof of NP-hardness will apply even in the simpler case of translation alone, without changes in scale.

This is identical to L&B's problem formulation, with two exceptions. First, they restricted themselves entirely to images related by translations. And secondly, they considered images in which figure and background were visible along only narrow, parallel slits. That is, they assumed that all of the image was occluded except for a set of parallel lines, which were completely colored in black and white. They also assumed that the tangent to the shape could be computed at the grey points separating white and black portions of these lines.

L&B then solved this problem by considering all possible combinations of matches between two black or two white line segments belonging to lines in different images. For each possible set of matches, they ran a linear program to find the feasible set of translations that match the lines in this way. Unfortunately, the number of ways of matching these lines is exponential in both n (the number of line segments in each image) and m (the number of images), leading to an algorithm that is both exponential in these values.

The L&B problem is also closely related to our own recent work on object recognition ([4, 11]). In this work we attempt to determine the pose of a known object from an image in which the object may be partially occluded. Both works share the assumption that no specific local features can be identified in an image and matched to a model, or to other images. Instead, one can identify subsets of the object in images, and must determine pose without specific correspondences of local features. In the case of our recognition work the object shape is completely known, which is comparable to supposing, in the current problem, that one of the images contains no occlusion. Our previous work focuses on showing that with this assumption, object pose can in many cases be uniquely determined from a single image containing considerable occlusion [5]. We also show that pose can be efficiently determined if we divide the object shape into convex parts, and we will use this insight to provide an efficient algorithm to solve the current problem when the sensed object is convex.

The current problem contains the additional complication that shape and relative pose must both be determined, using only partially occluded images. It is simplified in other ways, however, because it assumes that images are related by scaled translation only. In our previous recognition work, we allow for a wide range of transformations, including affine and perspective projections from a 3-D scene into a 2-D image.

Of course, there has been a great deal of work on the general problem of determining structure and motion from a sequence of images (see [9] for a recent review). However, that work generally makes quite different assumptions. Typically, that work considers 3-D world structures, which give rise to a much more complex problem. At the same time, that past work typically assumes either that motion and any occlusions that occur are small, or that local geometric features have been identified and tracked. Perhaps most relevant to the problem considered here is the motion tracking work of Huttenlocher, Noh, and Rucklidge [10]. They allow for only 2-D translations, and for significant amounts of occlusion, by matching shapes using a variation on the Hausdorff metric. This work still attacks a problem considerably different from ours in that it is still assumed that a significant portion of the shape is seen in both adjacent images, so that a direct comparison of shape can be made.

3 A solution that is polynomial in the image complexity

We will now present an algorithm that determines whether a feasible set of transformations exist that are consistent with the set of images that we have gathered, and that delimits this set of transformations and the object shapes that are consistent with them. A preliminary version of this

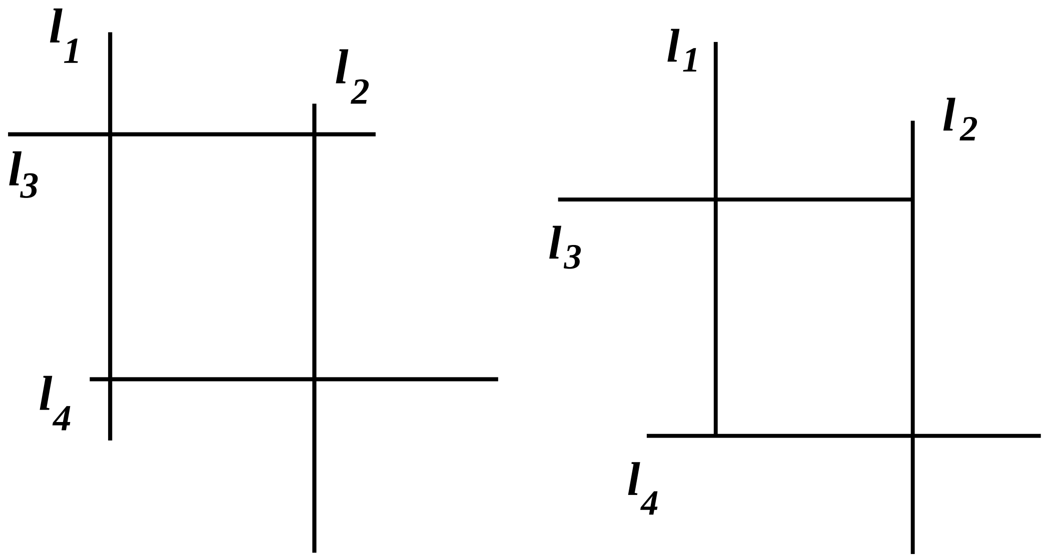

Figure 2: Two qualitatively similar translations for a simple example of the L&B problem. Here the "white" areas are only the remainder of the lines with black line segments. The remainder of the figure is "uncolored". On the right, the vertical lines are shifted upward and to the right until l_1 just touches l_4 and l_2 just touches l_3. Additional translation in these directions will result in a qualitatively different solution.

algorithm, which applied only to the L&B problem, appeared in [3].

We will describe the algorithm for the case where the images are related by a scaled translation. It is easily simplified, with reduced complexity, for the case of translation alone, and we will indicate how it may be extended to handle more complex transformations, at the cost of additional complexity. The algorithm runs in $O(n^{2(d+1)(m-1)})$ time, where n is the maximum number of polygon sides in any single image, m is the number of images, and d is the number of degrees of freedom in the allowed transformations. Therefore this algorithm is polynomial when m (and d) is fixed, and practical when $d\,m$ is small. In Section 4 we show that the problem is NP-hard when m is allowed to vary.

The first step in describing our algorithm is to demonstrate that the number of qualitatively different transformations relating two images is a low-order polynomial. Having done this, we show that one can enumerate these sets of transformations, and so discover which sets are feasible.

The goal of our algorithm is to describe the set of transformations that superimpose the sets of colored images so that they intersect only in compatible colors, subject, perhaps, to some limits on the range of allowable transformations. Consider first the case of only two images, in which we may assume without loss of generality that only the second image will be transformed. The relationship between the polygons describing the two images may be expressed entirely in terms of the lines and vertices that bound them. We will divide the set of feasible transformations into *qualitatively identical cells.* These will be sets of transformations within which none of the relationships between the lines or vertices bounding the polygons are altered. To form these cells, first imagine extending all of the line segments bounding the polygons in the two images into lines. We will call two transformations qualitatively identical when they both map the second image onto the first so that every vertex in each image is on the same side of every line in the other image. Now consider a valid transformation of the second image, one which places each black or grey vertex and line segment inside or on the boundary of a black polygon, and similarly places each white vertex or line segment inside a white polygon. Clearly any qualitatively identical transformation will place the vertices and lines of each image inside the same polygons of the other image.

A qualitatively identical cell in transformation space is delimited by the constraints that its transformations must map points to the appropriate side of lines, and lines so that they separate the same set of points. Let (x, y) denote a point in the first image, (x', y') a point in the second image, and let lines in the first and second image, respectively, be described by the equations:

$$Ax + By + C = 0 \tag{1}$$

and

$$A'x' + B'y' + C' = 0. \tag{2}$$

Further, we assume that the second image is transformed by a scaled translation in which we denote the translation by (t_x, t_y), and use s for the scale factor. In that case, the constraints that bound a cell of qualitatively identical transformations have the forms:

$$A(sx' + t_x) + B(sy' + t_y) + C > 0 \tag{3}$$

and

$$A'(\frac{x - t_x}{s}) + B'(\frac{y - t_y}{s}) + C' > 0. \tag{4}$$

Since $s > 0$ this last equation can be replaced by

$$A'x - A't_x + B'y - B't_y + sC' > 0. \tag{5}$$

Note that these constraints are linear in the unknowns, t_x, t_y, s. This means that a cell is the intersection of a set of half-spaces in the 3-D transformation space.

L&B made use of a similar decomposition of the possible transformations into qualitatively identical sets; Figure 2 gives an example. Recall that L&B restricted the allowable transformations to translations only. They pointed out that when two lines are constrained to intersect in two specific line segments of the same color, this places four linear constraints on the range of qualitatively identical translations, which are the translations that place the end point of each line segment on the appropriate side of the intersecting line. It follows that, in L&B's setting, every set of translations that are qualitatively the same is bounded by a set of linear constraints, and so is a convex subset of the space of all translations. Note that we can view L&B's case as that where the "polygons" are in fact just line segments: the boundary changes between white and black correspond to the vertices, and the edges are just the segments themselves. Using this interpretation, L&B's observations are included as a special case of our more general analysis.

We have shown that, in general, any qualitatively identical set of scaled translations is defined by a set of linear constraints, and thus such a set is convex subset of the set of possible transformations. We will now show how to enumerate all the vertices of these convex sets of transformation space, that is, all of the extreme points of the feasible scaled translations. Every vertex of every cell in the transformation space is formed by the intersection of three planes, that is, these vertices occur when three of the linear constraints on the transformations intersect. This occurs when transformations map three points (or lines) of the second image so that they lie on (or contain) corresponding lines (or points) in the first image. A match between three points and lines in the two images provides three linear equations with three unknowns, which will generally have a unique solution. (If these matches provide no or infinite numbers of solutions they may respectively either be ignored, or an arbitrary satisfying transformation may be considered.) If we consider translation alone (so that $d = 2$) then only two matches are needed. In general, there will be $O(n^{2d})$ such sets of matches. Each set of matches determines a transformation that may be on the boundary of a set of feasible transformations. To check this, we must see whether this transformation causes all lines and points to intersect like colors. We may check this in $O(n^2)$ time.

When there are more than two images, we can determine the possible transformations of each image relative to the first image. All possibilities are enumerated by generating similar matches between each additional image and the first, and then considering all possible combinations of these matches. This leads to $O(n^{2d(m-1)})$ combinations, each of which can be verified in $O(n^{2(m-1)})$ time. However, in practice we can gain greater efficiency by evaluating matches based on only a partial set of the images, and then extending only those matches that lead to valid translations.

We can contrast this algorithm with that proposed by L&B by noting that, instead of considering pairs of matches between each image, they considered all combinations of matches between every boundary point and every line in each image. Each combination was then checked for validity with linear programming. This led to a number of possible matches that was exponential in n. Our algorithm demonstrates that this search was unnecessarily redundant; most of these exponential number of combinations are inconsistent with a single transformation, and need never be considered.

Our algorithm is directly inspired by the work of Cass [6] (which is based on the work of Baird [2]). Cass considers the problem of finding a transformation that matches a maximum number of known model point features to a set of noisy image features known to lie inside convex polygons. Cass shows in this case that the number of qualitatively different transformations is polynomial, and can be efficiently explored. While we consider a different problem, we also rely on the insight that we can partition the space of transformations into a polynomial number of interesting cells.

Mount, Silverman and Wu [13] make use of a similar formulation of the combinatorial structure of the qualitatively different sets of transformations. They consider the sets of qualitatively different intersections possible between translating polygons to compute the possible areas of overlap between two polygons. This is related to our approach, since we seek transformations in which the white and black regions of two images do not overlap at all. Their paper contains interesting additional insights into this combinatorial structure.

4 The problem is NP-hard in the number of images

Our algorithm's computational complexity is not polynomial in the number of images. Unfortunately this may be unavoidable because, as we now show, the problem is in fact NP-hard. This is the case even for L&B's more constrained version of the problem:

Theorem 1: *The Lindenbaum-Bruckstein problem is NP-hard. The decision version of the problem—I.e., given a collection of images (sets of lines), deciding if there exists any object with which these images are consistent—is NP-complete.*

Remark: The problem is NP-complete even for some constant number of lines per image and constant number of distinct segments per line, so that only the number of images is variable.

Note that the number of lines per image, times the number of segments per line (i.e., the number of distinct black and white subsegments per slice, which is one plus the number of color changes between black and white), corresponds to the n parameter in the general case.

The remainder of this section is devoted to a proof of this result. We begin by noting that the fact that the decision problem is in NP follows immediately from our earlier remarks. That is, if there is a consistent image, it is always possible to verify this by "guessing" the corresponding extreme translations and then checking consistency. Both the size of the guess and the time needed to check consistency are polynomial.

We prove hardness by reduction to a variant of the well-known 3SAT problem [7, 8]. 3SAT is perhaps the canonical NP-complete problem. We define this problem again here, as a reminder and to establish notation:

Definition 1: *Let $\mathcal{V} = \{v_1, v_2, \ldots, v_V\}$ be a set of* variables. *A* literal *is either v_i or $\neg v_i$ for some $v_i \in \mathcal{V}$; let $\mathcal{L}$ be the set of literals. A* truth assignment *τ is a mapping from $\mathcal{L}$ to $\{true, false\}$ such that $\tau(\neg v_i) =$ not $\tau(v_i)$ for all v_i. A* clause *is a set of literals. A clause is* satisfied *by a truth assignment τ iff at least one literal in the clause is given value true by τ. In a* SAT problem *one is given a collection of clauses, and asked if there is any truth assignment that satisfies all the clauses. In a* 3SAT *problem each clause has at most 3 literals. A* 3-3SAT *problem is a* 3SAT *problem such that each variable appears in at most 3 clauses.*

Only the last definition here is nonstandard. As well as restricting the number of literals per clause, we also limit the number of times any single variable appears. However, one can easily show that 3-3SAT remains NP-complete, as follows. Consider any 3SAT problem, but suppose that some variable v_i is used $k > 3$ times. We can replace each occurrences of v_i by one of k distinct new variables, $v_{i,1}, v_{i,2}, \ldots, v_{i,k}$. If we then add clauses saying that $v_{i,1}$ is logically equivalent to $v_{i,2}$ (the two clauses $\{v_{i,1}, \neg v_{i,2}\}$ and $\{\neg v_{i,1}, v_{i,2}\}$ suffice), that $v_{i,2}$ is equivalent to $v_{i,3}$ (i.e., $\{v_{i,2}, \neg v_{i,3}\}$ and $\{\neg v_{i,2}, v_{i,3}\}$), and so on, then we clearly obtain a 3-3SAT problem that is satisfiable if and only if the original 3SAT problem has a solution.

The body of our proof shows how, if one is given a 3-3SAT problem with C clauses and V variables, one can construct an instance of the Lindenbaum-Bruckstein problem that is consistent if and only if the given 3-3SAT problem has a satisfying assignment. From this, it follows that the decision version of L&B's problem is also NP-complete.

The proof uses a somewhat involved construction which makes heavy use of a "lock-and-key" principle. A *lock* is a local collection of color changes in one image, matched by one or more corresponding *keys* in another. The two images can be consistently superimposed only when a key of the appropriate type is superimposed on the lock. Thus, by choosing the number and location of the keys, we can limit the possible intersections between two images to a discrete set of alternatives. This is the basic trick that allows us to reduce a combinatorial problem, such as 3-3SAT, to the Lindenbaum-Bruckstein problem, which is very geometric. Note that the (somewhat messy) details of our lock-and-key construction are not relevant to the basic strategy of the proof, so we defer these details until towards the end of this section.

Given a 3-3SAT problem, we must construct an instance of the Lindenbaum-Bruckstein which is consistent if and only if the given 3-3SAT problem has a solution. Note that a clause with 3 literals can be satisfied in one of seven possible ways. For instance, $\{v_i, v_j, \neg v_k\}$ is satisfied if v_i, v_j, v_k are true, or if v_i is false and v_j, v_k are true, etc.; in fact by any combination other than v_i and v_j being false and v_k being true. Thus any given 3SAT problem is consistent iff we can choose (1) for each variable, either *true* or *false*, and (2) for each clause, one of the seven ways of making the clause true, such that all of the choices we make are consistent with each other. This may seem like a somewhat redundant reformulation of 3SAT, but it turns out to be well-suited to our problem.

The main idea of the proof is to have one (horizontal) *selector* image for each variable, and one (vertical) selector image for each clause. We also use two *guides*: a vertical guide for variables and a horizontal guide for clauses. These guides are arranged as in Figure 3. Note that there is a (distinct) lock corresponding to each clause and variable on the appropriate guide; in each case these locks are distance 2 apart. A central lock $L0$ on the vertical guide, and corresponding key $K0$ on the horizontal guide, ensures that the two guides will have a fixed position relative to each other. We defer details of the guides' construction until the end, after we have shown how to define locks and keys. However we note here that, although the guides are rather complex combinations of images, each selector will be a single image consisting of a single line.

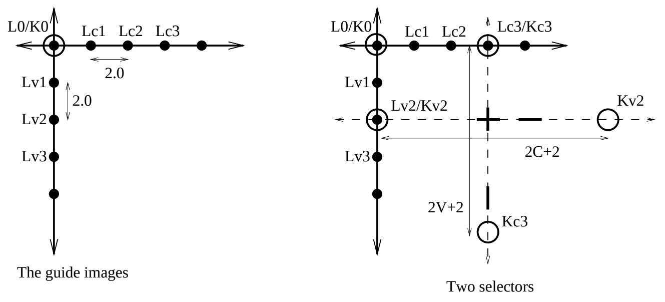

Figure 3: Guides and selectors. (Dashed segments are "white.")

Each variable selector will have two identical keys, distance $2C+2$ apart (where C is the number of clauses), each matching the corresponding lock on the variable guide. (I.e., in the notation of Figure 3, the selector for variable i will have two keys matching the lock Lvi.) It follows that the variable selector can be in one of two possible locations; intuitively, these will correspond to *true* and *false*. Similarly, each clause selector has seven copies of its key, spaced distance $2V+2$ apart, and so can be in one of seven distinct positions (intuitively, each corresponds to one of the seven ways the clause might be made true). See Figure 3 where two selectors are displayed.

The final step is to ensure that the selectors' locations are consistent if and only the original 3-3SAT problem has a solution. We do this by inserting black segments between the selectors' keys at appropriate points. Consider first a variable selector. After the second key (i.e., in Figure 3, the rightmost key), the line is uniformly white. But between the first key and the second the line is white except in locations corresponding to clauses in which the variable appears, at which points we insert a black segment of length 1. So, for instance, if v_i appears in the j'th clause, there is a black segment centered $2j$ to the right of the first key on v_i's selector. The variable selector shown in Figure 3 shows that variable 2 appears (possibly negated) in clauses 3 and 4.

Next we consider the clause selectors. Recall that each key on a clause selector is supposed to correspond to one way of making the clause true. Thus, it specifies a particular value (either *true* or *false*) for each of the three variables appearing in the clause. This assignment is reflected in the pattern on the selector between this key and the next: this gap should be uniformly white except that there are black segment(s) of length 1 at positions corresponding to any variable which both (1) appears in the clause being considered, and (2) is assigned *true* by the particular key being considered. For example, consider the clause $\{v_i, v_j, \neg v_k\}$, and suppose the first key corresponds to v_i being *false* and v_j, v_k being *true*. Thus, between the first key and the second should be solid white *except* at distances $2j$ and $2k$ (corresponding to v_j and v_k), at which points will be centered two black segments. The clause selector shown in Figure 3 shows that one way of making clause 3 true is to make both variables 2 and 4 true.

It is now easy to verify that the construction works. If the given 3-3SAT problem is satisfiable, then the corresponding alignment of selectors will be consistent and, conversely, if we can align the selectors consistently we can read off a satisfying assignment by looking at the variable selectors. That is, if a variable selector has its first key matching the corresponding lock we should regard that variable as being *true*, otherwise it is *false*. Consider any given arrangement of the variable selectors. A clause selector will be consistently translated if and only if all variables which the clause selector thinks should be true overlap black segments on the corresponding variables' selectors, implying that the variable selector agrees that the variable is true.

We complete the proof by discussing one way in which the "lock-and-key" pairs and the guides can be constructed. For reasons noted below, this is also a rather involved construction. A *key* will be any black segment in an image of length strictly less that 1; different length keys will fit different locks. Notice that our construction never uses segments of length < 1 in any other role, so nothing can be used as a key "by accident". It is also necessary to suppose that no two keys ever appear on a line within distance 1 or less of each other; our construction also clearly satisfies this.

To be concrete, we consider the case of a horizontal lock and a vertical key. A horizontal lock is part of an image consisting of 6 lines; see Figure 4. The 1st (i.e., the top) and 5th lines are distance 1 apart and are solid white, while the second line (distance ϵ below the top) is solid black. Here ϵ is a very small width that determines the precision with which the lock and the key must match. (In our construction, it suffices to take $\epsilon = 0.25/(C+V)$.) Lines 3 and 4 are a distance b and $b+3\epsilon$, respectively, below line 2, where $b < 1-4\epsilon$ is a parameter defining the lock (see below). Both lines 3 and 4 are white to the left, and black to the right, but line 4 becomes black slightly (in fact, ϵ) to the right of line 3. Let x denote the point where line 3 becomes black, and y the point where line 4 becomes black. Line 6, which is white except for a short black segment of length 0.5, should be ignored for the moment.

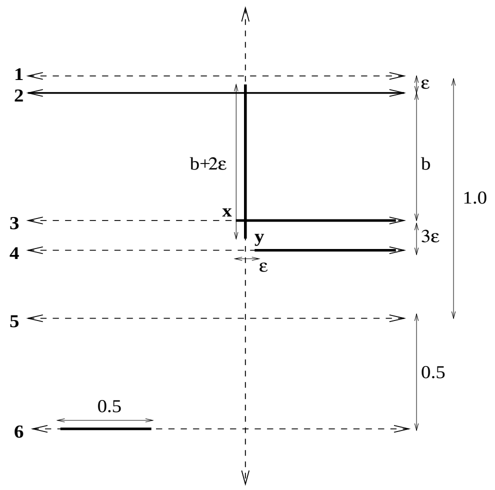

Figure 4: A horizontal "lock" and matching vertical "key"

Now consider any image which consists of a single vertical line, and whose only key(s) have width strictly less than $b+3\epsilon$. Such a line cannot cross the lock at any point to the right of position y, since it has no key long enough to cross more than two black lines at once. On the other hand, if all keys have width greater than $b + \epsilon$, then it must cross the lock at a point to the right of x, because there is no way that it can cross line 2 without also crossing line 3. It follows that a key of length $b + 2\epsilon$ must cross the lock between positions x and y. Thus, when the vertical image has a $b + 2\epsilon$ sized key (and no other key), we can fix the relative position of the images to within precision ϵ (which we can make as small as we wish). We can either fix the position completely or, if the vertical line contains several keys of length $b + 2\epsilon$, we can limit their relative position to a discrete set of alternatives.

Note that the construction so far provides a six-line image with a *single* lock on it. Let us call such a lock a *simple* b-lock. (I.e., we obtain different locks by varying the value of b.) However, the two guides in our construction appear to require many different locks, at differing positions. However, this is easy to arrange. Consider, for instance, the horizontal guide line, which requires $C + 1$ locks (i.e., $L0$ and a lock for each clause). For this, we consider a collection of separate simple $i/(C + 2)$-locks, for $i = 1 \ldots C + 1$. Our guide lines can be constructed by appropriately *superimposing* all these locks. That is, we need the top and bottom lines of all these simple locks to be (almost) on top of each other and furthermore that the "position" of the i'th lock be 2 units to the right of the $i - 1$'st lock. (Note, though, that we don't want *exact* superposition—the various locks would be inconsistent with each other—but rather just that they be very close to each other.) Of course, we can arrange the appropriate superposition with yet another lock! This is the purpose of the 6th line in simple locks; recall that this is always just a horizontal key of length 0.5. But putting this key in the appropriate positions in each horizontal lock, then adding a *single* vertical simple 0.5-lock, we can align all the horizontal locks as necessary for our construction. (We note that the construction of this vertical lock must be slightly different from that discussed so far, since it must not interfere with the first 5 lines of the horizontal locks that will cross it. The details of this easy variation are omitted.) The vertical guide line is, of course, constructed analogously.

Our discussion of locks-and-keys may seem somewhat complex than necessary. Although there are simpler constructions, ours has an important advantage: Each lock only uses a constant number of lines (i.e., 6), and only a constant number of color changes per line (never more than 2). Furthermore, our locks work even when appropriately superimposed, so we can construct our guides using many smaller images. In contrast, if we were to make each guide a single image, we would need the number of lines and/or the number of color changes to grow with C and V; consequently, the result would be much weaker. Aside from the images used in defining the guides, the only other images we use are the clause and variable selectors, and each of these only consists of a single line. Furthermore, the clause selectors clearly have only a finite number of color changes (although, as we have presented the construction here, clause selectors can have as many as 56). Note that each variable selector also has a constant number of color changes (in fact at most 10), but only because we considered the 3-3SAT problem in which any one variable appears in only a few clauses. This concludes the proof. **Q.E.D.**

5 Efficient solution for convex shapes

In the previous sections we introduced an algorithm for shape recovery that is polynomial in the complexity of the image, but then showed that even the simpler problem originally defined by L&B is NP-complete in the number of images. The latter result's proof required us to consider fairly odd shapes, so it therefore makes sense to consider placing some restrictions on shape in order to obtain efficient algorithms. In this section we consider the class of *convex* shapes. Convexity, together with another assumption about the configuration of the occlusions (explained below), allows us to solve the problem efficiently. Specifically, we can solve the problem by running a linear program with a set of constraints whose number is quadratic in both n and m. With the aid of a stronger assumption about the occlusions, we can reduce the number of constraints to be linear in n and m. We note that, given convexity of the image, this stronger assumption about the configurations of the occlusions will always hold in L&B's version of the problem.

Consider any single image, image i for instance, and let N_i denote the convex hull of all black regions in the image. Given the assumption that the unknown object is convex, the object clearly must include N_i. Furthermore, the points in N_i are the *only* points in image i that, on the basis of this image alone, can be deduced to definitely come from the image. Just as N_i includes all points that *must* be included in the object, we can also construct another region, which we call X_i, that includes all points that *might* be part of the object. One can find X_i as follows. Recall that the black regions in the image will be bounded by lines that are either black or grey, where grey corresponds to a boundary between black and white. Then consider the largest convex region which includes all of the grey lines as part of its boundary (see Figure 5). X_i is simply all non-white portions of this region. It follows that the shape of the unknown object must be "sandwiched" between N_i and X_i.

The second assumption we make in this section, in addition to the object's convexity, is that X_i is also convex. This will be the case when the grey lines delimit a fairly small region where the object may lie, and the occluder is large enough to cover this region. This occurs in Figure 5, for example. It is also the case whenever the white region is the union of a set of half-planes, which is always true for the L&B problem when the shape is convex (see Figure 6). To see this, note

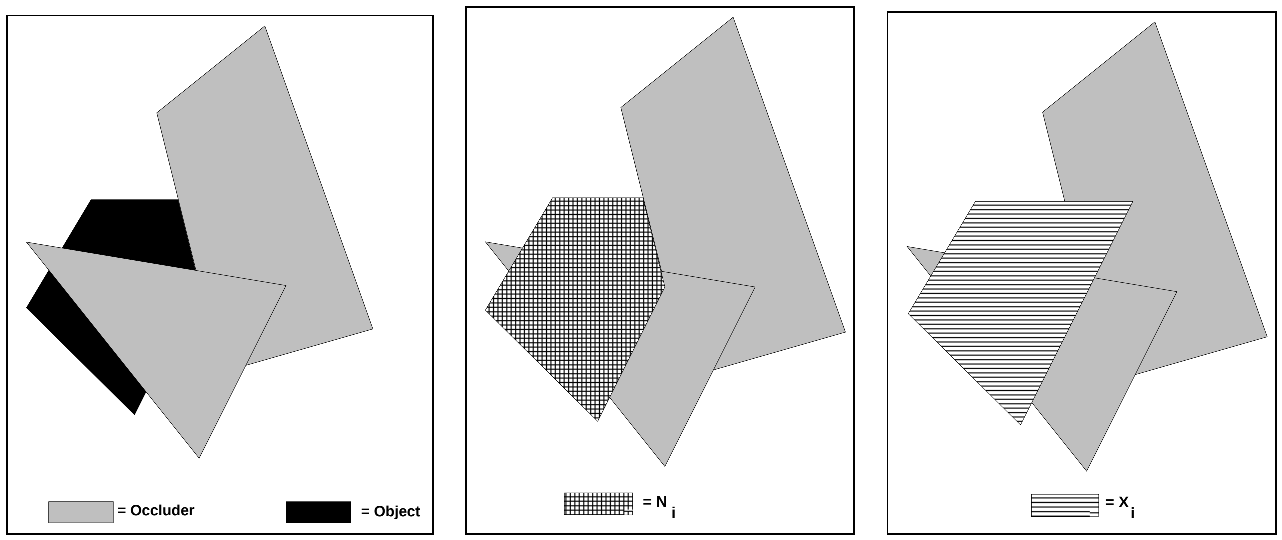

Figure 5: On the left, a scene with some occlusion. The middle figure shows the minimum region that may belong to a convex object in this figure. The right shows the maximum convex region.

that in this case each image consists of a set of lines that are white except for a single black line segment, together with (perhaps) some flanking lines on either side that are all white. Each black line segment is bounded by two points at which the tangent of the object shape is known. Each of these points, with its tangent, defines a line that bounds the largest possible region containing the shape. (Note that without some restriction, such as convexity, on the object's shape, the tangent information is essentially useless. This is why we did not consider this information in the preceding sections.) Additionally, the shape must lie entirely on a known side of any all-white line in the image.[1] Therefore, X_i in this case is simply the intersection of a set of half-planes, and so is convex (see Figure 6, right).

Given our assumptions, it is easy to see that all the images are consistent with each other if and only if we can transform them into a common reference frame, in which every N_i is contained in every X_j. This is because the intersection of the X_j contain no white, and so if every N_i is contained in this intersection, so is their collective convex hull, and so no known parts of the shape can overlap white.

To formalize this, let the vertices of N_i be $p^i_1, ..., p^i_n$ ($1 \leq i \leq m$), and let $p^i_k = (x^i_k, y^i_k)$. For notational convenience we assume that there are exactly n of these points in each image, although

[1]This is because if at least some portion of the shape is not visible in an image, that image may be ignored.

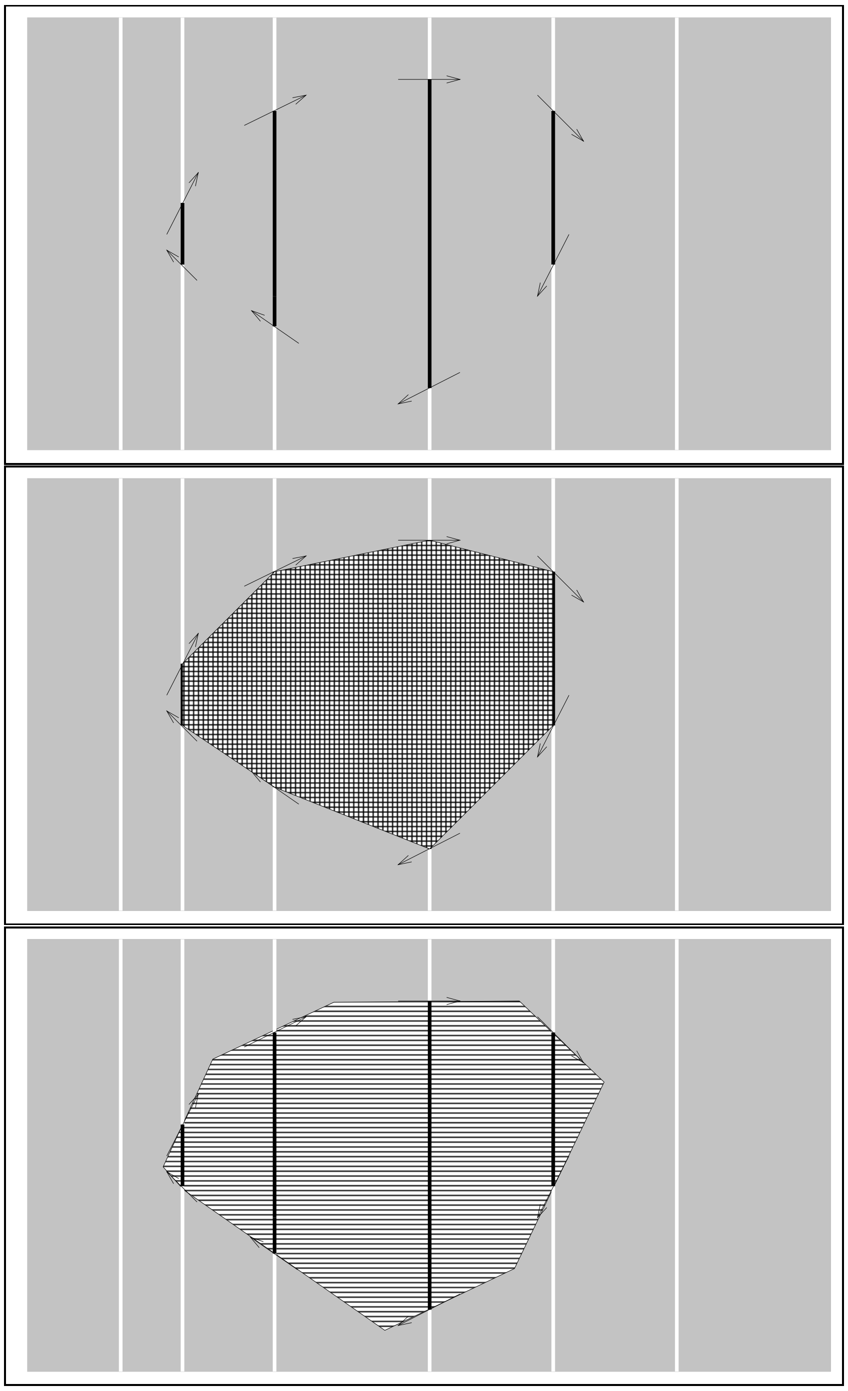

Figure 6: An example of the L&B problem, with a convex shape (top). Background is shown in grey, to make white lines visible. N_i (shown in middle) is the convex hull of the vertices. Every tangent or purely white line delimits a half-plane that must be all white. Together, all constraints delimit a convex X_i (bottom).

in fact there may be fewer. Next, we express all the lines bounding X_i using equations of the form $A^i_k x + B^i_k y + C^i_k = 0$ and assume, without loss of generality, that the shape lies in the positive half plane defined by the line. (I.e., that every point (x, y) in X_i satisfies $A^i_k x + B^i_k y + C^i_k \geq 0$.) Again, we simplify the presentation by assuming that each X_i is determined by exactly n lines.

Now consider first just two images, I_i and I_j. Denote the relative translation between the images by (u_{ij}, v_{ij}), and the scaling by s_{ij}. Then, for every point in image i and every line in image j, we may write two linear constraints of the form:

$$\begin{aligned} A^j_l(s_{ij}x^i_k + u_{ij}) + B^j_l(s_{ij}y^i_k + v_{ij}) + C^j_l &\geq 0 \\ A^i_k(x^j_l - u_{ij}) + B^i_k(y^j_l - v_{ij}) + s_{ij}C^i_k &\geq 0. \end{aligned} \tag{6}$$

Constraints of the first sort ensure that $T(N_i) \subseteq X_j$, where T indicates the transformation, while constraints of the second form require that $N_j \subseteq T(X_i) \equiv T^{-1}(N_j) \subseteq X_i$. These constraints ensure that the convex hull of all transformed black points are transformed to a position where they contain no white points. The point positions and the line equations are known in these constraints, while the scale and translation magnitudes, s_{ij}, u_{ij}, v_{ij} are unknown. It is possible to write all of these constraints as linear inequalities because both T and its inverse can be written as transformations that are linear in the same set of parameters.

These constraints can be written for every pairing of a point and line in two images, resulting in at most $2n^2$ constraints for the two images. We can find the scaling and translation parameters between I_i and I_j by solving this system of $2n^2$ linear inequalities using, e.g., linear programming. If we use Seidel's randomized algorithm ([14]) this linear program can be solved in expected time that is of the order of the number of constraints, which is $O(n^2)$.

The number of constraints can be further reduced for the special case where every vertex of N_i is grey, and we know its tangent. This is true in L&B's problem, where the endpoint of every line segment is the dividing point between the white and black parts of the line, and where it is assumed that the tangent of the shape is known at these points. Note that the angle of every line is invariant under scaled translation. Therefore, we can infer the order of these vertices on the boundary of the shape. The position of each vertex is only constrained by its neighbors. Therefore, the number of useful constraints between two images is twice the number of points in the two images, that is, only $2n$. Consequently, if the tangent directions in every image are given in order then we obtain an algorithm that is linear in the number of points in the two images. (Of course if the tangent directions are given unordered we will have to add to this the $O(n \log n)$ cost of sorting the angles.)

When considering more images, it must be true that all translations and scalings are consistent, i.e., $\forall i, j : 1 \leq i < j \leq m$

$$(u_{ij}, v_{ij}) = \sum_{k=i}^{j-1} (u_{k,k+1}, v_{k,k+1}), \qquad s_{i,j} = \prod_{k=i}^{j-1} s_{k,k+1} \tag{7}$$

Thus, with m images we have only $3m - 3$ unknowns. We now have $O(nm)$ points and lines, all pairs of which yield a linear constraint, for a total of $O(n^2m^2)$ constraints. Seidel's algorithm is no longer suitable, since it is practical only for linear programs with few variables. However, it is well known that linear programs can be solved in time that is polynomial in the number of constraints. In practice, the Simplex method typically takes time proportional to c^2d to solve a problem with c linear constraints in a d-dimensional space (see, e.g., [15]), which gives an overall run time of $O(n^4m^5)$. This will be practical when n and m are not too large.

For the L&B problem and related special cases where the vertices have known tangent directions, the number of constraints will be $O(nm)$. ($O(nm)$ vertices may need to be sorted in producing these constraints.) This leads to a total run time of $O(n^2m^3)$.

This algorithm is related to our own work on object recognition [4] and to Amenta's work on shape matching under the Hausdorff metric [1]. In our work we aligned 2-D models to 2-D images using correspondences between convex regions with no explicit correspondences between local features inside the regions. In that work we assumed that the shape of the object is given by the model, and that the image may be partly occluded. In our present terminology, this is equivalent to assuming that in the first image, $X_1 = N_1$, and in the second image, X_2 is unbounded. Given these assumptions, we need only enforce the constraint $T(N_2) \subseteq X_1$, since the constraint that $T^{-1}(N_1) \subseteq X_2$ is always true. This allows us to linearize our problem for a wider class of transformations, including similarity, affine and projective transformations. In that work we also show that constraints of the form $T(N_2) \subseteq N_1$ and of the form $T^{-1}(N_1) \subseteq N_2$ together cannot be linearized for these transformations; our argument in that paper also shows that our current problem of constraints of the forms $T(N_i) \subseteq X_j$ and $T^{-1}(N_j) \subseteq X_i$ cannot be simultaneously linearized for these more complex transformations.

Also, Amenta has shown how to efficiently find the scaled translation that minimizes the one-way Hausdorff distance between two polytopes. Amenta[1] shows that this problem can be solved using *convex* programming, a generalization of linear programming. In this problem, one polytope is translated and scaled so that it is placed inside another which has been dilated by a distance λ. This problem is easier than ours in that only one polytope is constrained to lie inside the other, so that constraints are formulated on the transformation, but not on its inverse. However, it is more difficult than our problem, in that the minimum dilation distance must be found for which the problem is still solvable.

6 Experiments

To illustrate our algorithms we have implemented and run them on three artificially generated shapes. In the first two sets of experiments we used two non-convex shapes, and so we applied the algorithm described in Section 3. In the first experiment images of a head-like shape were produced in the form allowed by L&B. Figure 7 shows three passes obtained for the head (the original shape is dotted). Five sensors at known positions were used to detect object and background segments along five lines. This resulted in 30, 20, and 14 boundary points in the three passes. Matching the lines in passes 1 and 2, for example, gave rise to 15000 translation cells (30*20*5*5=15000), of which only 12 were feasible. Similarly, matching passes 1 and 3 gave rise to 10500 cells, 13 of which were feasible, and matching passes 2 and 3 gave rise to 16500 cells, 5 of which were feasible. The figure also shows an overlay of the three passes obtained after matching all three passes. It can be seen that a good reproduction of the shape was obtained.

Next, we applied the same algorithm to a second, non-convex shape allowing for more general types of occlusions. Figure 8 shows three images of the shape with different occlusions and the shape recovered by aligning these three images. In this case too a good reproduction of the shape was obtained.

Finally, we tested our method with a convex, cupcake-like shape. As in the first experiment, the input in this case was provided in the form allowed by L&B. Figure 9 shows six passes obtained for this shape. This shape is convex, and so we used the algorithm described in Section 5. Again, five sensors at known positions detected object and background segments along five lines. Each sensor (denoted by tiny circles) reported the positions of two boundary points and their tangent direction, resulting in ten boundary points in each pass. Using a single linear program we aligned the six passes. The figure also shows an overlay of the six passes. Again, a good reproduction of the shape was obtained.

7 Conclusions

We consider the problem of recovering the shape of a heavily occluded object from a sequence of images. Lindenbaum and Bruckstein have proposed a specific example of this problem, in which the

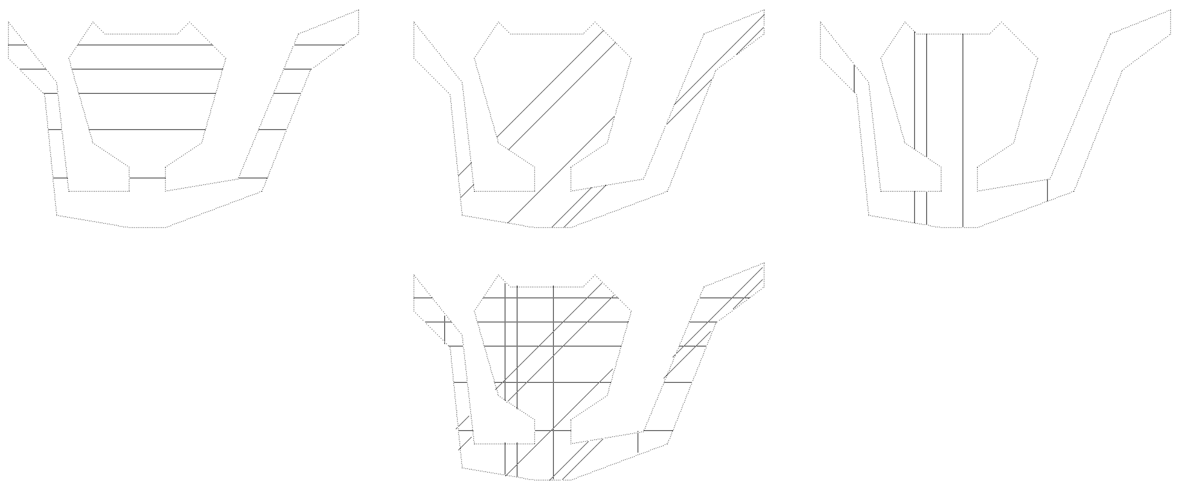

Figure 7: Three passes of a "head" shape (dotted lines) across a field of five sensors (top row). Solid lines show each sensor's output, and the result of aligning all measurements (bottom).

object is sensed through small apertures that model the receptive fields of insect sensors. We have shown that this problem is not tractable in general (assuming P $\neq$ NP). However, we can efficiently solve the Lindenbaum and Bruckstein problem when either the number of images combined at once is small, or if we know that the shape is convex. In fact, our algorithms apply to a significant generalization of the original problem, including cases where the 2-D images containing general patterns of figure, background and occlusion.

References

[1] Amenta, N., 1994, "Bounded boxes, Hausdorff distance, and a new proof of an interesting Helly-type theorem", *Proceedings of the 10th Annual ACM Symposium on Computational Geometry*: 340–347.

[2] Baird, H., 1985, *Model-Based Image Matching Using Location*, MIT Press, Cambridge.

[3] R. Basri, A. Grove, and D. Jacobs, 1996, "Efficient determination of shape from multiple images containing partial information," *13th International Conference on Pattern Recognition*, **A**:268–274.

[4] R. Basri and D. Jacobs, 1995, "Recognition using region correspondences," *International Conference on Computer Vision*:8–15.

[5] R. Basri, D. Jacobs, and O. Menadeva, "Pose estimation by matching regions: uniqueness and sensitivity to occlusion," *Forthcoming.*

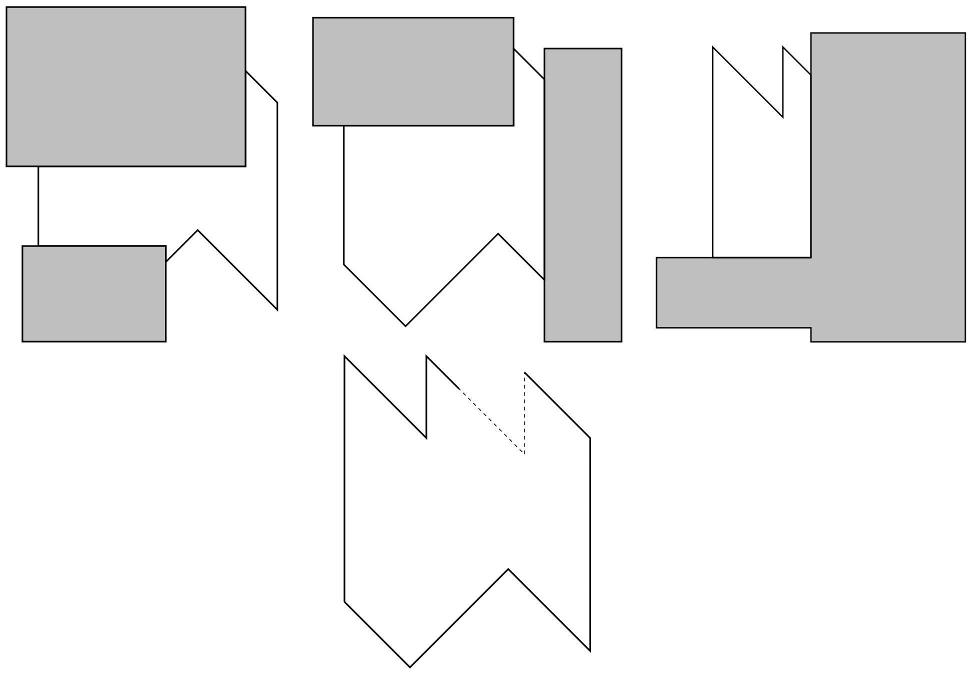

Figure 8: Three instances of a non-convex shape (solid lines) which are partly occluded (the occlusions are denoted by the gray regions, top row) and the recovered shape (bottom). The dashed line represents portions of the boundaries that are not determined by the three images.

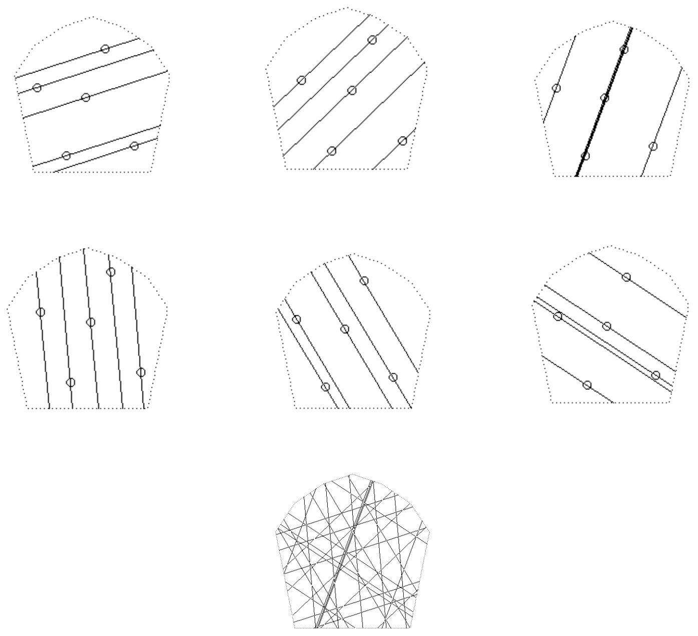

Figure 9: Six passes of a “cupcake” (top two rows), and the result of aligning all measurements, using the convex solution (bottom).

[6] Cass, T., 1992, "Polynomial time object recognition in the presence of clutter, occlusion and uncertainty," *Second European Conference on Computer Vision*: 834–842.

[7] S. A. Cook, 1971, "The complexity of theorem proving procedures," *Proceedings 3rd ACM Symposium on Theory of Computing*:151–158.

[8] M. Garey and D. S. Johnson, 1979, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. Freeman and Co., Publishers.

[9] Huang, T. and A. Netravali, 1994, "Motion and structure from feature correspondences: A review," *Proceedings of the IEEE*, **82**(2):252-268.

[10] Huttenlocher, D., J. Noh, and W. Rucklidge, 1993, "Tracking non-rigid objects in complex scenes," *4th International Conference on Computer Vision*: 93–101.

[11] Jacobs, D. and R. Basri, 1997, "3-D to 2-D recognition with regions." *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR-97): 547–553.*

[12] Lindenbaum, M. and A. Bruckstein, 1988, "Determining object shape from local velocity measurements," *Pattern Recognition*, **21**(6):591–606.

[13] Mount, D., Silverman, R., and Wu, A., 1996, "On the area of overlap of translated polygons," *Computer Vision and Image Understanding*, **64**(1):53–61.

[14] Seidel, R., 1990, "Linear programming and convex hulls made easy," *Proceedings of the Sixth Annual Symposium on Computational Geometry*, pp. 211-215.

[15] Strang, G., 1988, *Linear Algebra and its Applications*, Harcourt, Brace, Jovanavich, Publishers.