Placing Registration Marks

Anil S. Rao
Department of Computer Science
Utrecht University
3508 TB Utrecht, The Netherlands.

Kenneth Y. Goldberg
Institute of Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA 90089-0273, USA.

RUU-CS-93-21
June 1993

Utrecht University
Department of Computer Science
Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : + 31 - 30 - 531454

Placing Registration Marks

Anil S. Rao
Department of Computer Science
Utrecht University
3508 TB Utrecht, The Netherlands.

Kenneth Y. Goldberg
Institute of Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA 90089-0273, USA.

Technical Report RUU-CS-93-21
June 1993

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

ISSN: 0924–3275

Placing Registration Marks*

Anil S. Rao†
Department of Computer Science
Utrecht University
3508 TB Utrecht, The Netherlands.

Kenneth Y. Goldberg‡
Institute of Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA 90089-0273, USA.

Abstract

In industrial assembly, a registration mark can be placed on parts to aid a computer vision system in determining the position and orientation (pose) of parts. However, when sensor noise and limits on resolution introduce errors in the measured location of the registration mark, these errors can propagate into the measurement of part pose. In this paper we define the *Registration Mark Problem*: given an n-sided rigid planar polygonal part and a set of k poses for the part, locate a point on the surface of the part that maximizes the minimum distance between transformed points. A registration mark at this point will be maximally robust to sensor imperfections.

We give an $O(n \log n + k^4 \log k \log^* k)$ time algorithm to solve this planar problem using a result from Schwartz and Sharir [22] and demonstrate the algorithm using a commercial vision system. Our results extend to classes of curved planar parts and polyhedral parts.

1 Introduction

Determining the precise position and orientation (pose) of parts is a familiar problem in automatic assembly of electro-mechanical parts. Although some parts, such as resistors and ICs, are generally delivered in pre-packaged tapes or tubes, mechanical parts are often delivered in bags or boxes and must be individually oriented at the assembly site. One solution is to use mechanical constraints such as vibratory bowls or jigs to restrict the number of stable poses with a low-cost computer vision system to disambiguate among these poses.

Since computer vision systems are limited by pixel resolution and sensor noise, manufacturers can locate a *registration mark* on the part surface to reduce ambiguities. In this paper we show how to locate a single registration mark so as to optimally disambiguate between a finite set of poses for the part.

We formally define the *registration mark problem* as follows. Given a rigid planar part P, and a set of k planar geometric transforms $T_1, \ldots, T_k$, find a point $p_0 \in P$ that maximizes

$$\min_{i,j \in \{1,\ldots,k\}, i \neq j} d(T_i p, T_j p), \tag{1}$$

*This research was supported in part by NSF award IRI-9123747, ESPRIT Basic Research Action No. 6546 (project PROMotion), a Cooperative Agreement between the National Institute for Standards and Technology (NIST) and the Institute for Manufacturing and Automation Research (IMAR), and by an equipment grant from Adept Technology, Inc.

†Tel: 31-30-533922, Fax: 31-30-513791, Email: anil@cs.ruu.nl

‡Departments of Computer Science and Electrical Engineering–Systems, Tel: 1-213-740-9080, Email: goldberg@iris.usc.edu

where d is the Euclidean distance metric and $T_i p$ denotes the point obtained by transforming point p through T_i. See Fig. 1.

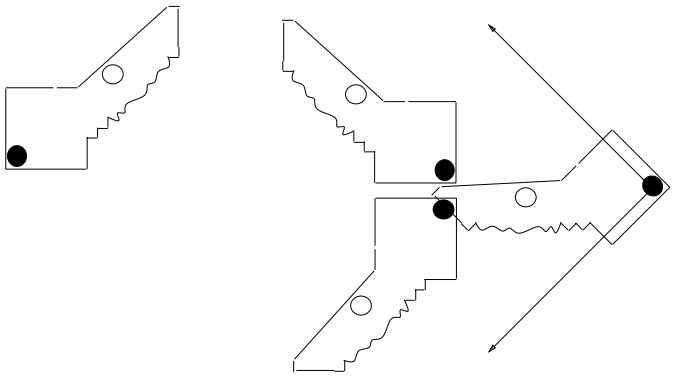

Figure 1: Due to mechanical constraints, a key-shaped part is constrained to one of the four poses shown above. We want to place a registration mark on the surface of the part so that we can distinguish between these poses using a computer vision system. Due to sensor noise, we want to maximize the separation between the transformed marks. The solid circles indicate a poor placement for the mark: the two middle poses could be confused The hollow circles indicate an optimal location for the mark.

For convenience we denote $d(T_i p, T_j p)$ as $d_{i,j}(x, y)$, where (x, y) are the Cartesian coordinates of p. In this notation the problem becomes:

$$\max_{(x,y)\in P} \quad \min_{i,j\in\{1,2,..,k\},i\neq j} d_{i,j}(x, y). \tag{2}$$

Since Euclidean distances are positive, the solution is unchanged if we replace $d_{i,j}$ by $d_{i,j}^2$, to simplify the computation. We call each $d_{i,j}^2$ a *distance function*. Note that there are $N = \frac{k(k-1)}{2}$ distance functions.

The function $M(x, y) = \min_{i\neq j} d_{ij}(x, y)$ is called the *lower envelope* of the N distance functions. If a registration mark is placed at a point (x, y), then $M(x, y)$ gives the minimum separation between the transformed marks. Therefore $M(x, y)$ can be termed the *registration mark distance* for the point (x, y).

A point (x_0, y_0) such that $\forall x, y \in P$, $M(x_0, y_0) \geq M(x, y)$ is called the *optimal registration mark point* for part P; and $M(x_0, y_0)$ the *optimal registration mark distance* for part P.

As an example, consider the four poses of a key-shaped part in Fig. 1. Each pose has a coordinate frame attached with it. Two locations for the registration marks are illustrated. One with solid circles and the other with hollow circles. Note that the latter is a better placement because the transformed marks are better separated. This can be verified by considering the lower envelope, M, for these transforms shown in Fig. 7,7 Note that $M(6.15, 6.83) > M(0, 0)$.

1.1 Related Work

In this paper we assume that the part can lie in one of a *finite* set of known poses. When the number of possible poses is uncountable, online estimation procedures are required [16, 18]. Note that in general, a part subject to the force of gravity will have only a finite number of stable poses.

Similarly, a parallel-jaw gripper (See Appendix A), or other mechanical constraints can be used to limit the number of stable part poses.

In this paper, we distinguish between stable poses using a camera to measure the position of a registration mark on the part. Grossman and Blasgen [13] considered a method for orienting parts by dropping them into the corner of a box which was vibrated (to reduce the effects of friction) until the part settled into a stable pose subject to the force of gravity. Stable poses were then distinguished using a tactile probe. Several other researchers such as [7, 17] took advantage of mechanical constraints to limit the complexity of pose estimation. In fact, it has been shown that a sequence of mechanical constraints can be sufficient to determine the pose of a part *without* sensors [12, 11, 19].

Our solution requires searching a planar map that represents the projection of $M(x, y) = \min_{i,j} d_{i,j}(x, y)$ on the xy plane. Such a map was studied by Schwartz and Sharir [22] who related its combinatorial complexity to upper-bounds on lengths of Davenport-Schinzel sequences [8] (See Appendix B). These bounds have been previously applied to problems in translational motion planning, hidden line/surface removal, and convex hulls in two and three dimensions [10, 22].

The results of this paper will appear in a special issue on Industrial Robotics of the IEEE Transactions on Industrial Electronics [21]. A preliminary version appeared in the 1993 IEEE International Conference on Robotics and Automation [20].

1.2 Trivial Special Cases

Two special cases of the problem yield trivial results:

1. All transforms are pure translations – any point on the part yields the same registration mark distance. See Fig. 2 (a).

2. All transforms are pure rotations about some point – the point on the part that is farthest from this point is optimal. In case of a polyhedral part, this would be one of the vertices of the part. See Fig. 2 (b).

The rest of the paper is devoted to cases where the transforms involve a combination of translation and rotation. To provide intuition behind our solution, we first consider a one-dimensional version of the problem.

2 Case of a rod

By a rod we understand a homogeneously one dimensional linear finite part. Without loss of generality, let the rod have length 1 and denote one of its endpoints as A. Any point on the rod can be characterized by its distance x from A, $0 \leq x \leq 1$. In this section we consider the problem of choosing the optimal position on the rod.

A planar transformation, $T_i = [a_i, \quad b_i, \quad \theta_i]$, relocates any point on the rod, x, to $(a_i + x\cos(\theta_i), b_i + x\sin(\theta_i))$. For a pair of transforms, T_i, T_j, the squared distance between transformed points is

$$d_{i,j}^2(x) = C_0 x^2 + C_1 x + C_2,$$

where $C_0 = 2(1 - \cos(\theta_j - \theta_i)), C_1 = ((a_i - a_j)(\cos(\theta_i) - \cos(\theta_j)) + (b_i - b_j)(\sin(\theta_i) - \sin(\theta_j))), C_2 = (a_i - a_j)^2 + (b_i - b_j)^2$.

Note that as a function of x, each distance function is a parabola that is concave up (since its second derivative is positive) and that k transforms define $N = \frac{k(k-1)}{2}$ such functions. The

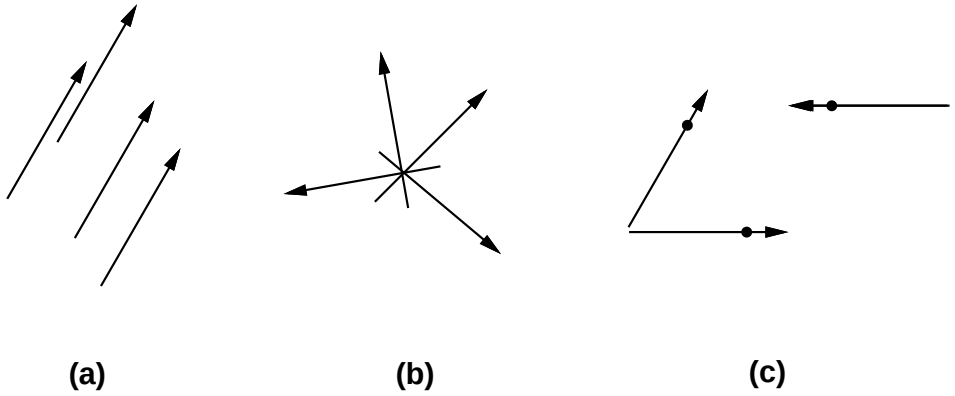

Figure 2: Three sets of transforms on a one dimensional "rod". In (a) all the transforms are translations and therefore any point on the part is equivalent to any other for placing a registration mark. In (b) the transforms are all rotations about a point q. The point on the part farthest from q is the best choice for placing the mark. In (c) the transforms include both translations and rotations. If the length of the part is 1, the best placement of the mark in case (c) is at a distance 0.82 from the tail as indicated with a solid circle.

registration mark problem is solved by computing the lower envelope of all these parabolas in the range $[0,1]$ and finding the maximum point in this envelope. See Fig. 3.

Consider the N parabolas (distance functions), indexed $d_1, d_2, \ldots, d_N$ for convenience of notation. The idea is to always maintain a partition of the space $[0,1]$ as a collection of internally disjoint intervals, and associated with each interval of the partition, the index of the parabola (among all the parabolas so far considered) forming the lower envelope throughout that interval. That is, after processing i parabolas, $[0,1]$ should be partitioned as $[0 = a_0, a_1, \ldots, a_{n(i)} = 1]$, and there should be an associated array A such that, for each $j \in [1,2,\ldots,n(i)]$, and each $q \in [1,2,\ldots,N]$, and for all $x \in [a_{j-1}, a_j]$, $d_{A(j)}(x) \leq d_q(x)$.

Lemma 1 *The complexity of the partition (or lower envelope), i.e. the number of intervals in it, $n(i)$, after processing i parabolas is at most $2i-1$.*

Proof See [5]. □

For the first parabola, there is only one interval; its lower envelope is simply itself restricted to $[0,1]$. As we proceed, the naive approach is to intersect each new parabola with each of the $n(i)$ indexed $A(1),\ldots,A(n(i))$ to check if any update in the intervals is required. Assuming computing the intersection between two parabolas takes $O(1)$ time, the processing time for the $(i+1)$th parabola is $O(i)$, since $n(i) = O(i)$. Thus, the complete running time to process N parabolas is $O(N^2)$. After the phase of computing the lower envelope, simply traverse it from left to right to find its maximum. This phase takes $O(N)$ time.

Computing the lower envelope of the N parabolas can be done more efficiently as follows. Sort all the N parabolas left to right on basis of the x-coordinate of their lowest point. Compute the lower envelope $M(L)$ for the left $N/2$ parabolas, L, and similarly but separately compute $M(R)$ for the right $N/2$ parabolas, R. Each of these sub-envelopes will have complexities $O(N)$.

Lemma 2 *The two lower envelopes $M(L), M(R)$ can be merged in $O(N)$ time.*

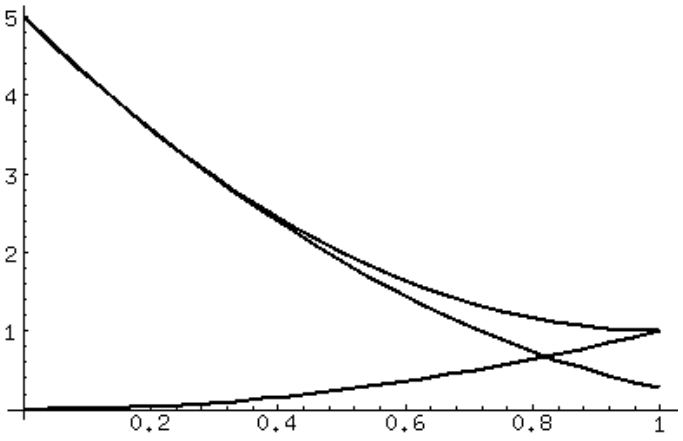

Figure 3: The three distance functions (parabolas) arising from the three transforms on a rod shown in Fig. 2 (c). The lower envelope is the point-wise minimum of the three parabolas. The maximum in the lower envelope occurs at $x = 0.82$ which gives the optimal placement of the registration mark.

Proof: The basic idea is to compute the intersection point I between $M(L)$ and $M(R)$. This can be done in $O(N)$ time by scanning the sequence of parabolas segments in $M(L)$ from left to right until a parabola segment is found that intersects $M(R)$. Now the merged lower envelope is simply $M(L)$ to the left of I and $M(R)$ to the right of it. □

Theorem 1 *The registration mark problem can be solved in $O(k^2 \log k)$ time for k transforms on a one-dimensional part.*

Proof: From Lemmas 1 and 2 we get that the time complexity for finding the lower envelope for N transforms, $t(N)$ is such that

$$t(N) = 2t(N/2) + O(N),$$

which gives $t(N) = O(N \log N)$.

Once the lower envelope, represented as a partition of $[0, 1]$, is computed, scan the parabolic segments in it from left to right and maintain the maximum. Since the lower envelope has complexity only $O(N)$, this step can be accomplished in $O(N)$ time. Overall time required is $O(N \log N) = O(k^2 \log k)$, since $N = k(k-1)/2$. □

3 Case of a polygon

Now we extend the one-dimensional approach to solving the two-dimensional problem: finding the optimal location for a registration mark on a planar polygon. First, we make two observations:

The Decomposition principle: The optimal location for a registration mark on a union of parts is equal to the optimal location on one of its components.

The Containment principle: If a part P has an optimal registration mark at p_0, then any subset of P containing p_0 will have the same registration mark.

A 2-D analog to the analysis in the previous section would be to characterize an arbitrary point within a polygon as (x, y), with appropriate bounding linear constraints on x, y. The distance function $d^2_{i,j}(x, y)$ between transforms can be seen to be of the form:

$$d^2_{i,j}(x, y) = Ax^2 + By^2 + Cxy + Dx + Ey + F,$$

However, in the case of planar rigid transforms, $A = B$ and $C = 0$ which enables us to use efficient techniques from Computational Algebra to find solutions [22]. Define functions f, g as two distance functions.

$$f(x, y) = d^2_{i,j}(x, y) = Ax^2 + Ay^2 + Dx + Ey + F,$$

and

$$g(x, y) = d^2_{l,m}(x, y) = ax^2 + ay^2 + dx + ey + f.$$

Their intersection is the 2-D curve $f = g$ which is a circle or its degenerate forms: a straight line or a single point [15] because the coefficients of x^2, y^2 match and there is no xy term.[1]

We use some notation from [22] to simplify further discussion. Recall that the function $M(x, y)$ is the lower envelope of the N distance functions. M induces a partition on the polygon (or in general, over the part shape) into maximally connected regions such that over any one such region, M is attained by a single distance function $d_{i,j}$. The boundary of such a region consists of either

[1] Otherwise the intersection can be a parabola, hyperbola, or ellipse.

arcs from the boundary of the part or arcs along which M is attained by at least two of the distance functions. Let M^* denote the planar map obtained from this partitioning. M^* is called the minimization diagram of the N distance functions. By the (combinatorial) complexity of M, we understand the number of regions in M^*. We next prove two lemmas using results from [22].

Lemma 3 *The combinatorial complexity of M, the lower envelope of N distance functions, is $O(N^2)$.*

Proof: Let f, g, h be three arbitrary distance functions. If the plane curve $f = g$ is simple, connected and partitions the plane into two disjoint regions; and the equations $f = g = h$ have at most two roots, then the combinatorial complexity of M, the lower envelope, for such distance functions is $O(N^2)$, as proven by Schwartz and Sharir [22]. These conditions are satisfied by our distance functions $d^2_{i,j}$ as shown below.

From the generic expressions for $f = d^2_{i,j}, g = d^2_{l,m}$ given in equations above, we find that the curve $f = g$ is of the form

$$(A - a)(x^2 + y^2) + (D - d)x + (E - e)y + (F - f) = 0$$

which is a circle (if $A \neq a$) or straight line (if $A = a$), both of which are simple, connected and partition the plane into two disjoint regions. Now consider the curves $f = g$ and $g = h$. Each of these is a circle or straight line, and so both these curves can intersect in at most 2 points as required. □

Lemma 4 *M^*, the minimization diagram (partition) of the N distance functions over the entire plane, can be computed in $O(N^2 \log N \log^* N))$ time.*

Proof: Again, let f, g, h be three arbitrary distance functions. [22] shows that M^* over the entire plane can be computed in $O(N\lambda_{s+2}(N) \log N)$ time, where s is the maximum number of roots of $f = g = h$ and λ is the maximum length of the Davenport-Schinzel sequence.

In our case, $s = 2$ as shown in the proof of Lemma 3. To determine bounds on $\lambda_4(N)$, we refer to Szemeredi [23] who showed that for any $r \geq 3$, $\lambda_r(N) = O(N \log^* N)$. Although there exist better bounds on $\lambda_4(N)$ (Agarwal *et. al* proves that $\lambda_4(N) = \Theta(N2^{\alpha(N)})$, where α is the almost-constant functional inverse of the Ackermann's function [2]), we use the $O(N \log^* N)$ bound given by Szemeredi.

Thus, the time complexity of computing the minimization diagram, M^*, over the entire plane is $O(N\lambda_4(N) \log N)$, which is at most $O(N^2 \log N \log^* N)$. □

Let a *basic computation step* refer to either the evaluation of one of the distance functions at some point, or determining the maximum of some distance function f along some portion of the contour $f = g$.

Lemma 5 *Each basic computation step can be performed in $O(1)$ time.*

Proof: Evaluation of a distance function at some point can clearly be done in $O(1)$ time. Consider finding the maximum of a distance function, say $f = d^2_{i,j}$ along a circular or linear arc segment δ, a straight line segment or an arc of the circle $f = g$, where $g = d^2_{l,m}$ is another distance function. Let p, q denote the end-points of the segment δ. To compute the maximum of f along the segment δ, we first need to compute the local maxima of f along the entire curve $f = g$. We describe this in the next paragraph and also show that the number of local maxima will be a constant. It is

clear that we only need to evaluate the function f at p, q and those local maxima that lie in δ to compute the required quantity.

Now we consider the problem of computing the maximum of f along the entire circle (or line) $f = g$ in $O(1)$ time (*i.e.* an exact closed form solution rather than a numerical technique). This is done by the method of Lagrange Multipliers. That is, by solving the following system in x, y, γ :

$$\nabla(f - \gamma(f - g)) = 0 \quad \text{and} \quad f - g = 0. \tag{3}$$

The first equation gives us x and y in terms of γ alone. Specifically, $x = \frac{x_1\gamma + x_0}{b_1\gamma + b_0}$ and $y = \frac{y_1\gamma + y_0}{b_1\gamma + b_0}$. Substitute these forms for x, y into the second equation $f = g$. This results in a quadratic in γ which can be solved exactly by analytic closed-form techniques. Obtain x, y from the real solutions to γ and examine if they are local maxima or minima. □

Theorem 2 *The registration mark problem can be solved in $O(n \log n + k^4 \log k \log^* k))$ time, given k transforms on an n-sided polygonal part.*

Proof: The first step is to compute the minimization diagram M^* for the part, given the k transforms and N distance functions. By Lemma 4, this can be done in $O(k^4 \log k \log^* k))$ time.

M^* has a combinatorial complexity (number of regions/edges/vertices) of $O(N^2)$ by Lemma 3 over the entire plane. Thus, over the polygonal part P having n edges, it could have a combinatorial complexity of $\Theta(N^2 n)$. However, we now prove that we can compute the maximum in the lower envelope over the part in $O(n + N^2)$ time by showing that only $O(n + N^2)$ basic computation steps are necessary to compute the maximum rather than the naive approach of evaluating the lower envelope at all the possibly $\Theta(N^2 n)$ points of intersection between P and $M*$. This technique has been previously applied in [3] in the context of finding the Hausdorff distance between two polygons.

Notice that because of the non-negativity of the coefficients of x^2 and y^2 in the distance functions, the maximum of the lower envelope over the part P can lie only on the boundary of P or along some arc of the minimization diagram $M*$. See Fig. 5 for an example minimization diagram.

First we notice that there are $O(N^2)$ arcs in $M*$. Consider some arbitrary arc. Let it be bounded by some curve $f = g$, where f, g are distance functions. We saw in the proof of Lemma 5 that f (or g) has a constant number of local maxima along the circle $f = g$. Compute these local maxima and break the arc at each local maxima. This process increases the total number of arcs by at most a constant and each arc now has the property that if p, q are the two end-points of the arc, the lower envelope function $M(x, y)$ along this arc from p to q decreases first and then increases and therefore attains its maximum at one of the two end-points of the arc. Let us refer to this as the "arc property." Evaluate the lower envelope at each of these arc end points (basically involves evaluating a known distance function). Notice the total number of basic computation steps performed is $O(N^2)$.

Next, we locate the n vertices of the polygon in $M*$. This can be done in $O(n \log N)$ time with $O(N^2 \log N)$ of preprocessing.[2] Evaluate the lower envelope function at each of these n vertices. This requires n basic computation steps. We also need to test each of the $O(N^2)$ vertices of $M*$ for inclusion in part P. This can be likewise done in $O(N^2 \log n)$ time assuming $O(n \log n)$ preprocessing. Now evaluate M at all the vertices of $M*$ that lie inside or on P. This involves $O(N^2)$ basic computation steps.

To avoid confusion, we refer to segments in $M*$ as "arcs" and those of P as "edges." Now consider an arc. P could intersect this arc in $O(n)$ points, but because of the arc property mentioned

[2] Point location is a standard problem in computational geometry. See [9].

above, only the *first* and the *last* intersection points are of any interest. See Fig. 4. Computing these first and last intersection points can be done by sweeping from p, q, respectively. The maximum of the lower envelope along the straight line segments AC and BD have now to be computed.

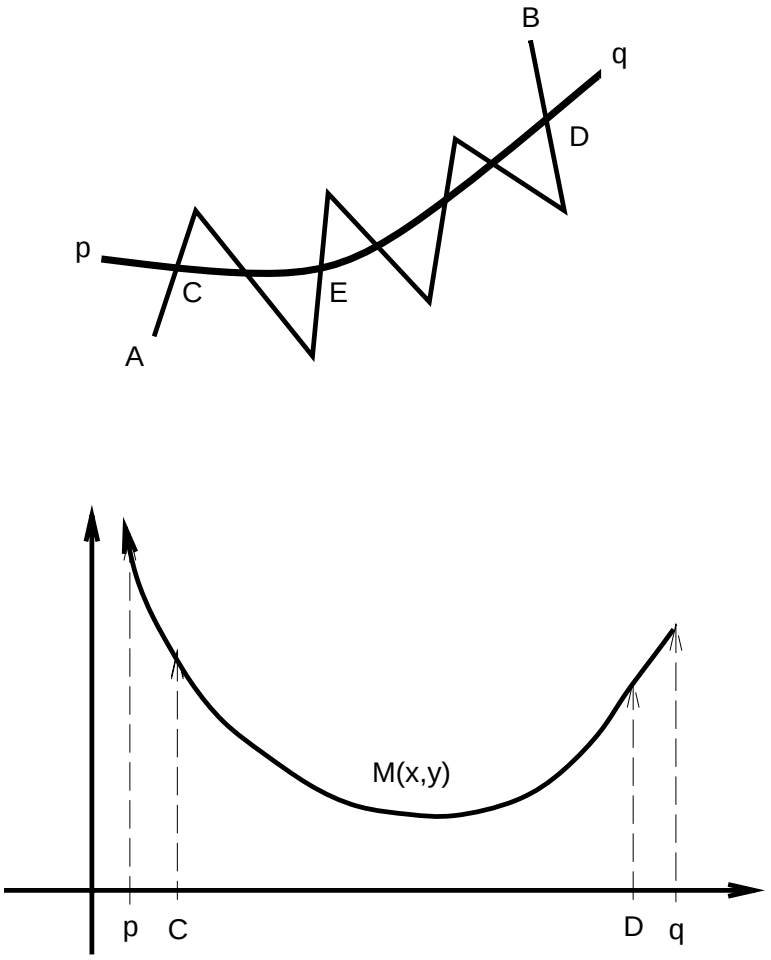

Figure 4: Only two intersections of P with an arc are relevant. **(top)** pq is an arc intersecting the polygon P at possibly several points. **(bottom)** Due to the arc property, the behavior of the lower envelope from p to q is such that it first decreases and then increases. Therefore in finding the maximum of the lower envelope over P, only the first and last intersection points C, D (more precisely, the portions AC and BD) are relevant. Other interior intersection points such as E can be ignored.

Thus, in all we have $O(n + N^2)$ basic operations. Along with this we need $O(n \log N)$ time for locating the n vertices of P in $M*$ assuming $O(N^2 \log N)$ of preprocessing, and $O(N^2 \log n)$ time assuming $O(n \log n)$ of preprocessing for testing inclusion of the vertices of $M*$ in P. Adding all this to the cost of computing $M*$, and noticing that $O(N^2 \log n + n \log N) = O(n \log n + N^2 \log N)$, the total computational cost for solving the registration mark problem over the polygon P is $O(n \log n + N^2 \log N \log^* N)$ time as claimed. □

3.1 An alternative formulation

Based on a preliminary version of this paper, Yan-Bin Jia and Mike Erdmann of CMU pointed out that the distance functions $d_{i,j}(x,y)$ could be formulated as the multiplicative weighted Euclidean distance between two points: (x,y) and $p_{i,j}$, *i.e.*

$$d_{i,j}(x,y) = w_{i,j}\ \mathrm{dist}(\ (x,y)\ ,\ p_{i,j}\ ), \tag{4}$$

where $w_{i,j}$ and $p_{i,j}$ are constants that depend on the given parameters of the matrices T_i, T_j:

$$w_{ij} = |2\sin((\theta_i - \theta_j)/2)|;$$

$$p_{ij} = \frac{1}{w_{ij}^2} \begin{pmatrix} \cos(\theta_i) - \cos(\theta_j) & \sin(\theta_i) - \sin(\theta_j) \\ -(\sin(\theta_i) - \sin(\theta_j)) & \cos(\theta_i) - \cos(\theta_j) \end{pmatrix} \begin{pmatrix} x_i - x_j \\ y_i - y_j \end{pmatrix}.$$

With this formulation we could construct the multiplicative weighted Voronoi diagram (MWVD) of the $k(k-1)/2$ points $p_{i,j}$ with associated weight $w_{i,j}$. Imagine point crystals placed at sites $p_{i,j}$ and growing at the rates of $1/w_{i,j} > 0$. The edge set of the MWVD is the locus of points where these grown crystals meet. Thus we obtain a partition of the plane into regions around each site $p_{i,j}$, each point in the region being closest (in the weighted) sense to its $p_{i,j}$ than any other. The MWVD for $k(k-1)/2$ points has combinatorial complexity $\Theta(k^4)$ and is computable in that time up to a constant factor [6].

The optimal location for the registration mark is then a point on some Voronoi edge, interior to the part, that maximizes the weighted distance to its site. This formulation thus eliminates a factor of $\log^* k$ from the 2nd term of the time complexity given previously.

4 An Example

We used *Mathematica* Version 2.0 [24] to solve the equations in the following example. Consider the following three planar transforms ($k = 3$) given as $[x,\ \ y,\ \ \theta_z]$: $T_1 = [0.5,\ \ 0.5,\ \ 0], T_2 = [-0.5,\ \ 0.5,\ \ \pi/2], T_3 = [-0.25,\ \ -11,\ \ \pi/4 - \pi/36]$ (*i.e.* θ_4 is 40°). Below, we indicate squared distance functions as $d_{i,j}$ (instead of $d_{i,j}^2$).

$$d_{1,2}(x,y) = 1 + 2x + 2y + 2(x^2 + y^2)$$

Numerical approximations for the others are:

$$d_{1,3}(x,y) = 132.8 - 14.4x + 6.3y + 0.5(x^2 + y^2)$$

$$d_{2,3}(x,y) = 132.3 + 8.6x - 17.4y + 0.7(x^2 + y^2)$$

By $d_{i,j,l,m}$ we imply $d_{i,j} - d_{l,m}$.

$$d_{1,2,1,3}(x,y) = -131.8 + 16.4x - 4.3y + 1.5(x^2 + y^2)$$

$$d_{1,2,2,3}(x,y) = -131.3 - 6.6x + 19.4y + 1.3(x^2 + y^2)$$

$$d_{1,3,2,3}(x,y) = 0.5 - 23.0x + 23.8y - 0.2(x^2 + y^2)$$

Solving, we find two points where $d_{1,3,2,3} = d_{1,2,2,3} = d_{1,2,1,3}$:

$$\{(4.8, 5.1), (-10.3, -8.2).\}$$

See Fig.s 5, 6(a), (b).

4.1 Experiments

To illustrate how the choice of registration mark can affect a machine vision system in the presence of noise, we generated the following example of a part in one of 4 possible transformations. The first three transforms are taken from the previous section.

The 4th transform we used in our experiments is three dimensional (involves a flip out of the plane): $T_4 = [x, \ y, \ z, \ \theta_x, \ \theta_y, \ \theta_z] = [-2.0, \ 25.0, \ 0, \ \pi, \ \pi]$. Three-dimensional transforms on polyhedral parts can be accommodated by solving the problem for each face of the part and combining solutions when the point is guaranteed to be visible from all transformations (for example, consider a hole through the part). Squared distance functions $d^2_{i,j}(x, y)$ corresponding to a pair of 3D transforms on a face are of the form $Ax^2 + By^2 + Cxy + Dx + Ey + F$, with possibly C non-zero and $A \neq B$. In general, two such surfaces intersect over non-circular conical curves, any two of which clearly intersect in up to $s = 4$ points. Therefore, by results of [22], M^* now has a slightly increased complexity of $O(N\lambda_{s+2}(N)) = O(N\lambda_6(N))$. However, this is also upper bounded by $O(N^2 \log^* N)$ [23], and so Theorem 2 holds for k 3D transforms as well.

For these four poses, we computed the six associated distance functions and their lower envelope on the plane, and then found the maximum within a bounding circle. In part coordinates, the optimal registration mark was located at $(6.15, 6.83)$. *Mathematica 2.0* required under a minute to find this solution on a Sun IPX workstation. For comparison, we also chose the origin of the part coordinate frame as a suboptimal location for the registration mark. We then constructed a "key"-shaped part lying wholly inside the bounding circle and containing both locations. See Fig.s 7(a), (b).

To suggest the type of noise that may be encountered by a machine vision system under non-ideal conditions, we generated blurred images of the part with each registration mark. We first generated two laser printer pictures of the part in each of the 4 possible poses, one with the registration mark at the origin (suboptimal) and the other with the registration mark located optimally.

A Sony Camera was then used to capture a 512×512 grey-level image of each printout. Image-Calc processing software from SRI was used to window the images down to 450×250 pixels. These Images A and B are shown in Fig.s 8(a) and 8(b). Next we performed Gaussian smoothing using a 3×3 filter to blur each image, followed by expanding grey-levels in the range 100-150 to full scale to enhance contrast. The resulting images, A$'$ and B$'$, are shown in Fig.s 9(a) and 9(b).

We now compare Fig.s 9(a) and 9(b). In Image A$'$, two of the suboptimal registration marks are difficult to distinguish. Thus the associated poses of the part would be difficult to disambiguate using an automated vision system. In comparison, each registration mark is easily distinguishable in Image B$'$. This illustrates how a careful placement of the registration mark can enhance pose recognition.

5 Discussion

In this paper we formally defined a problem related to pose estimation by machine vision: given a finite number of poses for the part, place a registration mark on the part so as to maximally separate the transformed poses. Our solution is efficient and borrows from previously known work in computational algebra.

This problem is a special case of *continuous maximin*. The general form of which can be solved with numerical optimization techniques [4, 14] that guess a solution and iteratively improve it. Such approaches may not converge on the optimum value and it is difficult to characterize the computational complexity of these approaches.

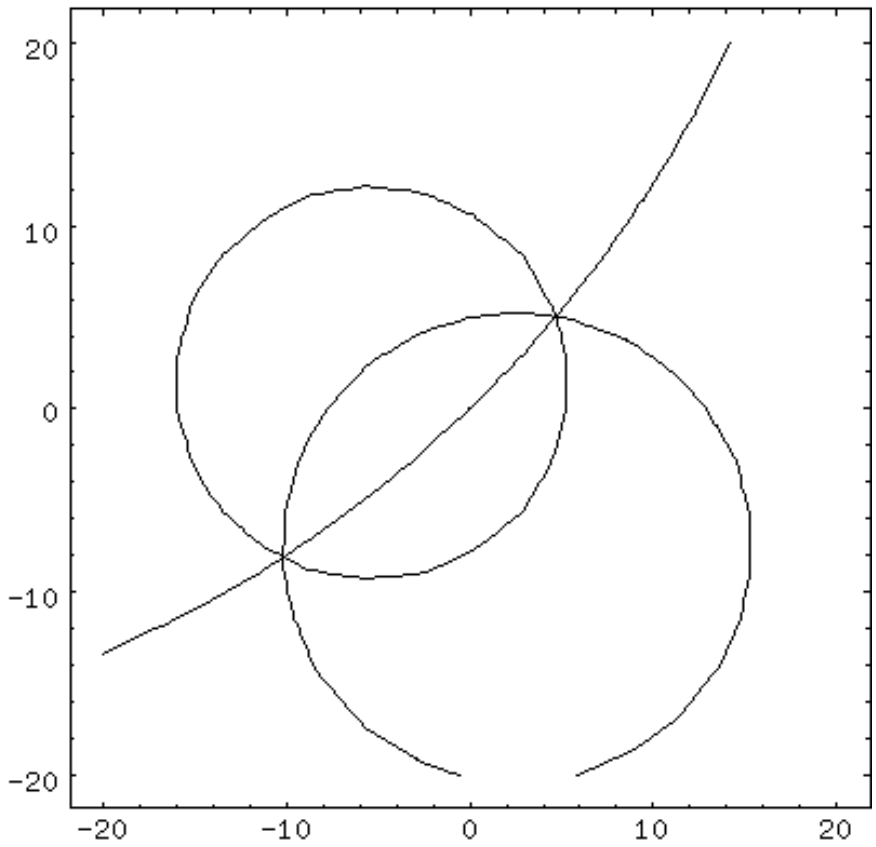

Figure 5: Minimization diagram for the three example transforms of Section 4. The three curves (circles) $d_{1,2,1,3} = 0, d_{1,2,2,3} = 0$, and $d_{1,3,2,3} = 0$ mutually intersect at the two points $(4.789, 5.126)$ and $(-10.298, -8.197)$ (The circles are shown incompletely). These two points are local maxima in the lower envelope of $d_{1,2}, d_{1,3}, d_{2,3}$ and can therefore be considered good registration mark points (see next two figures). Any connected region of this minimization diagram is bounded by circular arcs. There are three regions in this minimization diagram. Each connected region is shown associated with exactly one of the distance functions f that achieves the lower envelope M over the complete region. For example, the region shaded vertically is associated with the distance function $d^2_{1,3}$. Throughout points in this region, $d_{1,3} \leq \min\{d_{1,2}, d_{2,3}\}$, equality being achieved only at the boundary of the region.

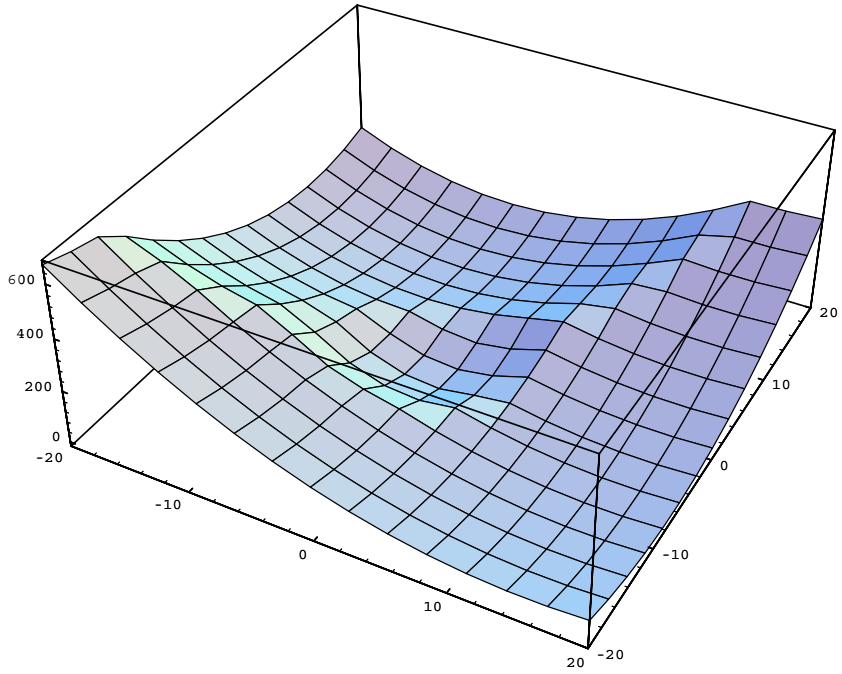

Figure 6: **(a)** Example of a lower envelope. Refer to the three transforms given in Section 4. The lower envelope of $d_{1,2}, d_{1,3}, d_{2,3}$ over the square $x, y \in [-20, 20]$. However, the local maxima at $(4.789, 5.126)$ and $(-10.298, -8.197)$ are not quite discernible. See **(b)** for a zoom in.

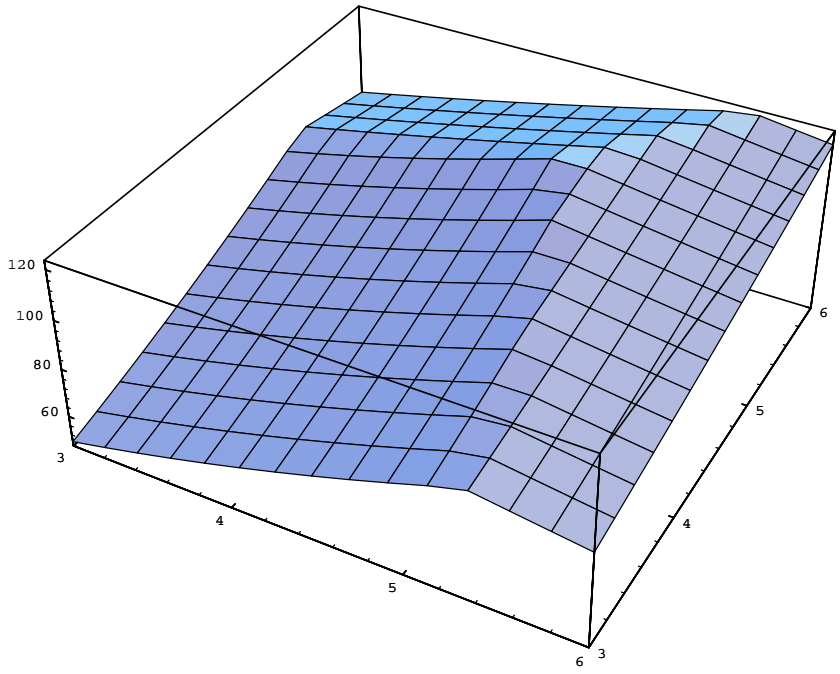

Figure 6: **(b)** Refer to the Fig. 6(a). To observe the local maximum at $(4.789, 5.126)$, we zoom in and observe the lower envelope of $d_{1,2}, d_{1,3}, d_{2,3}$ over the smaller square $x, y \in [3, 6]$. Thus, for the set of three transforms we began with, $(4.789, 5.126)$ is a good registration mark point. So is the other local maximum $(-10.298, -8.197)$ which isn't shown here.

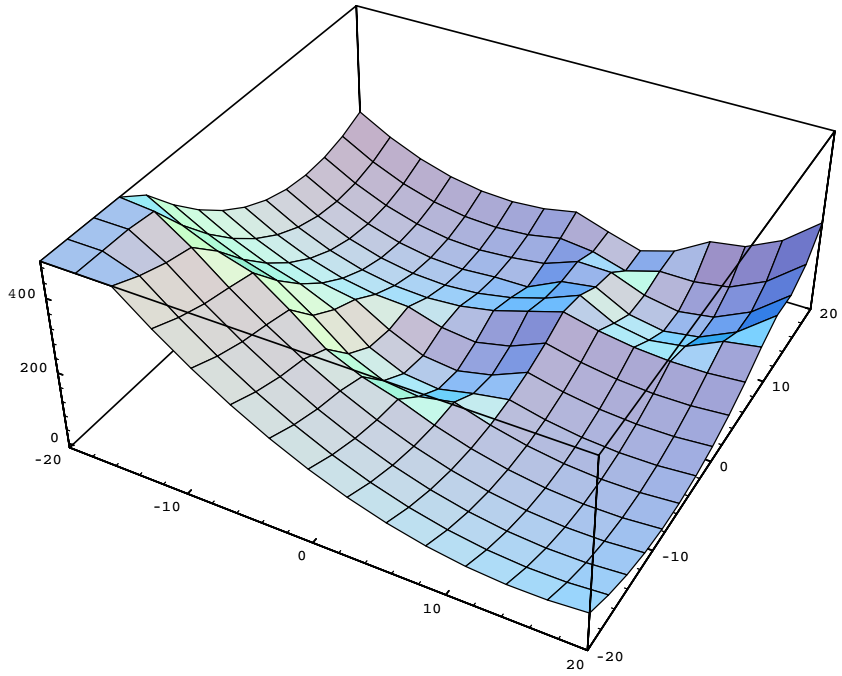

Figure 7: **(a)** The $k = 4$ transforms case: The lower envelope M for the $4 \times 3/2 = 6$ distance functions over the square $x, y \in [-20, 20]$. Compare the complexity of this lower envelope versus the one given in Fig. 6 for the case of three transforms.

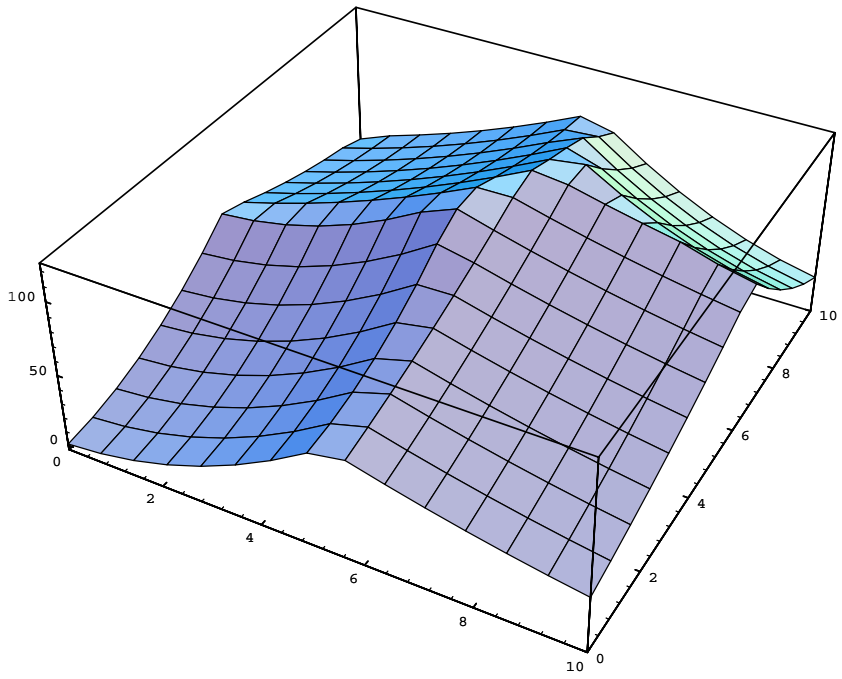

Figure 7: **(b)** Zooming into the lower envelope M of the previous figure to observe portion over the square $x, y \in [0, 10]$. Notice that $(0, 0)$ is clearly a bad choice for placing the mark (because the registration mark distance corresponding to $(0, 0)$, $M(0, 0)$, is small) while $(6.15, 6.83)$ is good ($M(6.15, 6.83)$ is high).

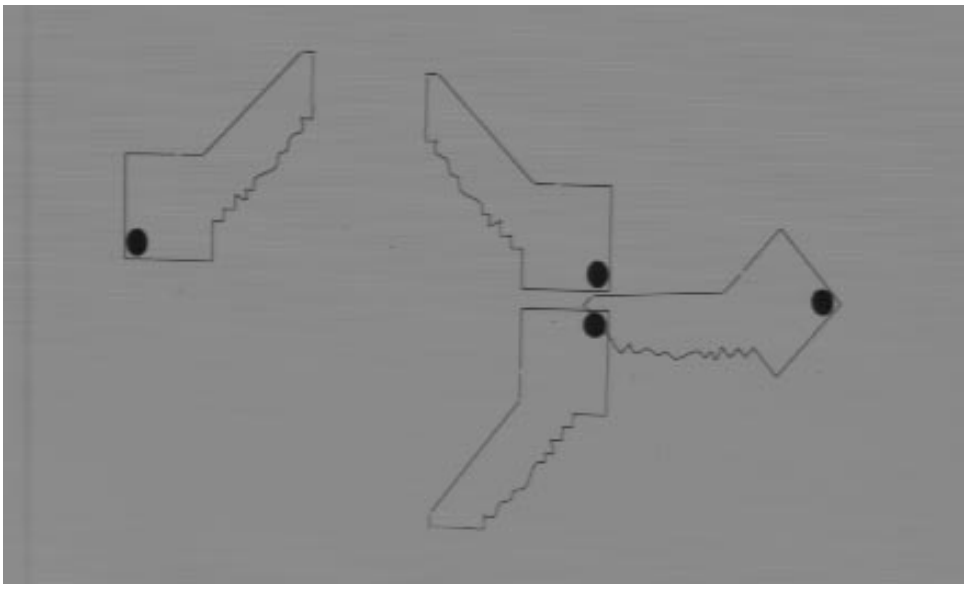

Figure 8: **(a)** Image A: Windowed image of the four keys with bad registration marks.

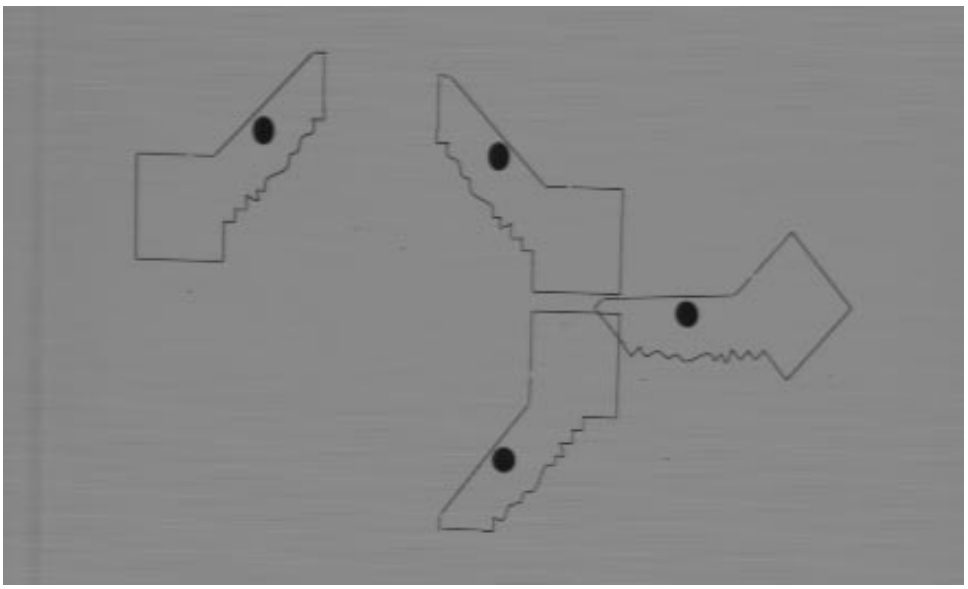

Figure 8: **(b)** Image B: Windowed image of the four keys with good registration marks.

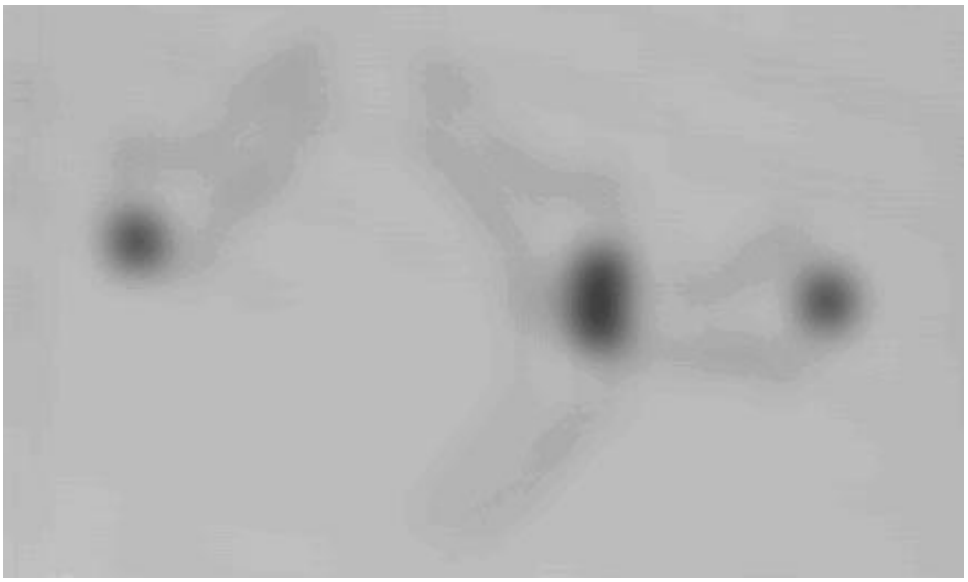

Figure 9: **(a)** Image A′: Image A after corrupting with Gaussian noise. Note that two of the registration marks are indistinguishable due to sensor noise.

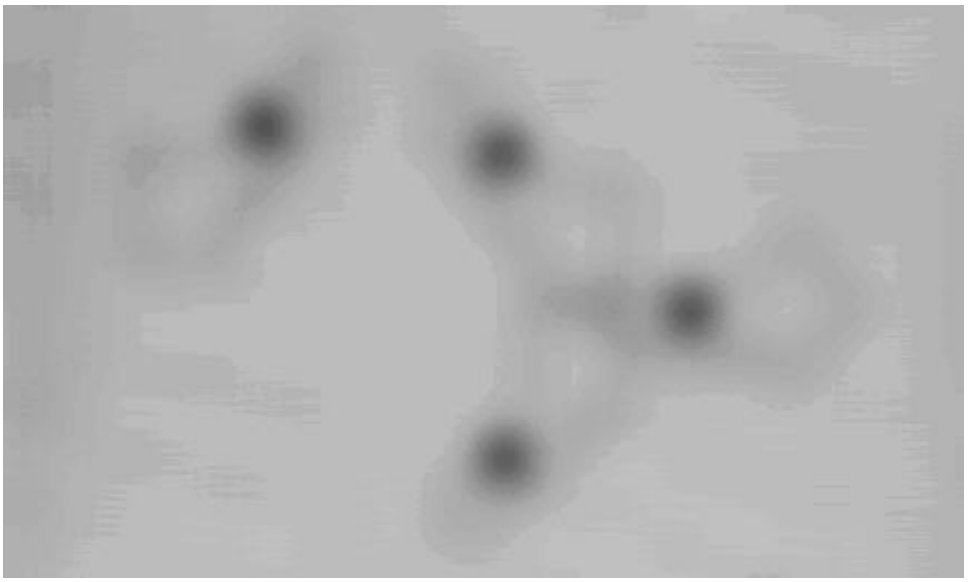

Figure 9: **(b)** Image B′: Image B after corrupting with Gaussian noise. All marks remain distinct in the presence of noise.

If noise sensitivity is known in advance, we can define a "separation threshold" for transformed registration marks. That is, we can compute the lower envelope over the entire plane and use the threshold to slice the envelope at a particular height, d_s, given by the sensitivity. This will define corresponding regions of the plane where a registration mark can be safely placed. Such regions could be useful if we later want to move the mark due to functional requirements.

Another alternative is the brute-force solution that searches over a large number of candidate points and explicitly finding the minimum distance between each point. For m points and k transforms, this would require $O(mk^2)$ time and does not guarantee an optimal placement of the mark. Our algorithm does find an optimal placement and does not depend on the sampling density.

We can also use our approach to place registration marks on non-polygonal planar parts. $M*$ has circular-arc boundaries; thus parts with linear and circular boundaries can be easily handled without any additional complexity. Other conic boundaries will introduce quartics into the γ in Lemma 5. Quartics also have exact analytic solutions so that each basic computation step will still run in constant time. Part boundaries of higher degree will introduce higher degree terms into γ; thus numerical techniques would have to be employed in the search for a solution. In such a case, the Decomposition Principle could also be exploited to decompose a part with a complicated boundary into several simpler shapes (say triangles) and compute the registration mark position for each one.[3]

A private communication from Pankaj Agarwal [1] indicates that perhaps the minimization diagram for the entire plane can be computed $O(k^4 \log k)$ time (cuts off a $\log^* k$ factor from the result presented in Theorem 2). This will be investigated in future research and if true, would match the complexity using MW Voronoi Diagrams (Section 3.1).

Acknowledgments

We thank Babu Narayanan for numerous fruitful discussions; Otfried Schwarzkopf and Mark de Berg for discussions on the proof of Theorem 2; Pankaj Agarwal for his comments on the complexity of the minimization diagram; Yang Chen for his help with the experiments; Jia Yan-Bin and Michael Erdmann for suggesting the use of Multiplicative Weighted Voronoi Diagrams; and Mark Overmars and Viktor Prasanna for their support.

The Appendix

A Mechanical Constraints on Part Pose

As stated in the Introduction, we can restrict the number of stable part poses using mechanical constraints such as gravity. Here, we give an example using a two-axis frictionless parallel-jaw gripper as shown in Fig. 10(a). The pair of jaws L_1, H_1 form one gripper axis and L_2, H_2 the other. Imagine the two-axis gripper functioning as two single-axis grippers operating in tandem, L_1, H_1 closing first and then L_2, H_2, closing towards a central point C until further squeezing would deform the part (*i.e.* this can be accomplished with a pneumatic gripper and an arrangement of bearings). It is straightforward to compute all the stable poses of the part within this gripper. For a polygonal part with n edges, there can be at most $4n$ such poses. See Fig. 10(b) for the stable poses for the part shape shown in (a).

[3] However, triangulating an n-gon and solving the registration mark problem for each triangle results in $O(nk^4 \log k \log^* k)$ time complexity which is more than that obtained in Theorem 2 by considering the polygon as a whole. Thus, decomposition should only be done for complicated part shapes.

Figure 10: **(a):** A two-axis gripper poised over a trapezoidal part P. There are two pairs of jaws, one formed by H_1, L_1 and the other H_2, L_2. Both pairs of jaws close in towards a central point, C. The pair H_1, L_1 closes in as far as possibly (without deforming the part) before H_2, L_2 does. **(b):** Possible final poses for P after a single complete grasp.

B Davenport-Schinzel sequences

A sequence of integers $U = (u_1, \ldots, u_m)$ is called an (n, s) Davenport-Schinzel sequence [8] if

1. $1 \leq u_i \leq n$,
2. For every $i < m$ we have $u_i \neq u_{i+1}$, and
3. There do not exist $s+2$ indices $1 \leq i_1 < i_2 < \ldots < i_{s+2} \leq m$ such that $u_{i_1} = u_{i_3} = u_{i_5} = \cdots = a$, $u_{i_2} = u_{i_4} = u_{i_6} = \cdots = b$, and $a \neq b$.

By $\lambda_s(n)$ is understood the maximum length, m, of such an (n, s) Davenport-Schinzel sequence. The problem of estimating $\lambda_s(n)$ has been studied repeatedly beginning with the discoverers and Szemeredi [23] and ending[4] with Agarwal, Sharir, and Shor [2]. Szemeredi showed the generic bound $\lambda_s(n) = O(n \log^* n)$ for any $s \geq 3$. It is fairly easily shown that $\lambda_1(n) = n$ and $\lambda_2(n) = 2n - 1$. These are the results used in this paper. Better bounds for $\lambda_s(n)$ may be found in [2].

References

[1] P. Agarwal. Time complexity of computing lower envelopes. Private Communication, April 1992.

[2] P. Agarwal, M. Sharir, and P. Shor. Sharp upper and lower bounds on the length of Davenport-Schinzel sequences. *Journal of Combinatorial Theory, Series A*, 52:228–274, 1989.

[4] for the present!

[3] H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes. In *Annual Symposium on Computational Geometry*, pages 186–193. ACM, 1991.

[4] M. Aoki. *Introduction to Optimization techniques.* Macmillan, Inc., New York, 1971.

[5] M. J. Atallah. Dynamic computational geomeetry. *Comp. Math. with Applications*, 11:1171–1181, 1985.

[6] F. Aurenhammer and H. Edelsbrunner. An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern Recognition*, 17(2):251–257, 1984.

[7] A. D. Christiansen. *Automatic Acquisition of Task Theories for Robotic Manipulation.* PhD thesis, School of Computer Science, Carnegie Mellon University, 1992.

[8] H. Davenport and A. Schinzel. A combinatorial problem connected wiwth differential equations. *American Journal of Mathematics*, 87:684–694, 1965.

[9] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1987.

[10] H. Edelsbrunner, L. J. Guibas, and M. Sharir. The upper-envelope of piecewise linear functions: Algorithms and Applications. *Discrete and Computational Geometry*, 4:311–336, 1989.

[11] M. Erdmann, M. Mason, and G. Vanecek. Mechanical parts orienting: the case of a polyhedron resting on a table. In *International conference on Robotics and Automation* (ICRA), pages 360–365, Sacramento, CA, April 1991. IEEE.

[12] M. A. Erdmann and M. T. Mason. An exploration of sensorless manipulation. In *IEEE International Conference on Robotics and Automation*, 1986.

[13] D. D. Grossman and M. W. Blasgen. Orienting mechanical parts by computer-controlled manipulator. IEEE *Transactions on Systems, Man, and Cybernetics*, pages 561–565, 1975.

[14] G. Hadley. *Nonlinear and Dynamic Programming.* Addison-Wesley, Reading, Mass., 1964.

[15] W. H. Hart. *Analytic Geometry and Calculus.* Heath, Lexington, Mass., 1963. page 323.

[16] B. K. P. Horn. *Robot Vision.* MIT-Press/McGraw-Hill, 1986.

[17] D. J. Kriegman. Computing stable poses of piecewise smooth objects. *CVGIP: Image Understanding*, 55(2), March 1992.

[18] R. Nevatia. *Machine Perception.* Prentice-Hall, Inc., New Jersey, 1982.

[19] A. S. Rao and K. Y. Goldberg. Orienting generalized polygonal parts. In *International conference on Robotics and Automation* (ICRA), Nice, France, May 1992. IEEE.

[20] A. S. Rao and K. Y. Goldberg. Placing registration marks. In *International conference on Robotics and Automation* (ICRA), pages I: 161–167, Atlanta, GA, May 1993. IEEE.

[21] A. S. Rao and K. Y. Goldberg. Placing registration marks. IEEE *Transactions on Industrial Electronics*, February 1994. (to appear in a special section on Robotics Technology).

[22] J. T. Schwartz and M. Sharir. On the two-dimensional Davenport-Schinzel problem. *Journal of Symbolic Computation*, 10:371–393, 1990.

[23] E. Szemeredi. On a problem by Davenport and Schinzel. *Acta Arithmetica*, 25:213–224, 1974.

[24] S. Wolfram. *Mathematica*. Addison-Wesley, 2nd edition, 1991.