Pattern Recognition Letters 7 (1988) 51–57
North-Holland

An application of linear programming to object location in r dimensions

L.W. BRINN
The University of Michigan-Dearborn, Dearborn, MI 48128, USA

Received January 1987
Revised 6 May 1987

Abstract: This report describes an approach to locating an object in an r-dimensional scene. The location problem is reduced to a problem in computational geometry, then solved by linear programming methods.

Key words: Linear programming, object location, polygon matching.

1. Introduction

A central problem in computer vision is that of determining whether a given object appears in a scene and, if the object does appear, determining its location.

In the following we state a restricted version of the problem and outline a uniform method of solution applicable to r dimensional location spaces, where $r \geq 2$. We then extend both problem and solution in directions suggested by the method.

2. The basic location question

Suppose that we are given an object O. We wish to determine whether the object appears in a given scene S. The scene is assumed to be a bounded region in r dimensional Euclidean space ($\mathbb{R}^r$). The most common situation occurs when $r=2$ or $r=3$. However, higher dimensions are also of interest. In particular, in tracking the motion of a body or the progress of a process, we may introduce a coordinate corresponding to time. Additional coordinates might correspond to conditions under which observations were made (for example, lighting conditions under which a picture was taken, temperature of an observed object or process) or to the equipment used (for example, type of camera, type of film).

For the purposes of our discussion, the object O is represented by the values of some feature function f at certain distinguished points $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_n$ on O. We let Q be the polyhedron in $\mathbb{R}^r$ with vertices $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_n$ and refer to Q as the query polyhedron. Clearly, Q represents the original object O with an accuracy which depends on both the number and location of the points $\bar{p}_i$ and on the choice of the feature function f. Both the points and the function f are assumed to have been preselected.

We assume that there is a metric d defined for feature values and that a tolerance $t=(t_1, t_2, \ldots, t_n)$ has been given. For each i, we let R_i be the set of points p in S for which $d(f(p), f(\bar{p}_i)) \leq t_i$ and refer to the R_i as the object regions. Intuitively, it is reasonable to consider that object O appears in scene S if we can find points $p_1, p_2, \ldots, p_n$ in S such that the points p_i are related spatially as the points $\bar{p}_i$, and, for each i, the feature value at p_i is within the scalar tolerance t_i of the feature value at $\bar{p}_i$. With this motivation, we say that O has been found in S to within a tolerance t iff there exists a translation T of the query polyhedron Q such that, for each i, $T(\bar{p}_i) \in R_i$. The original location problem has been reduced to a problem in computational geometry. This approach is due to W. Grosky.

We first consider the restricted problem in which each region R_i is a nonempty closed convex polyhedron. If we assume that each object polyhedron R_i is given as the intersection of m_i closed half-spaces, we show that we can determine whether the required translation exists in time $O(m)$, where $m = m_1 + m_2 + \cdots + m_n$.

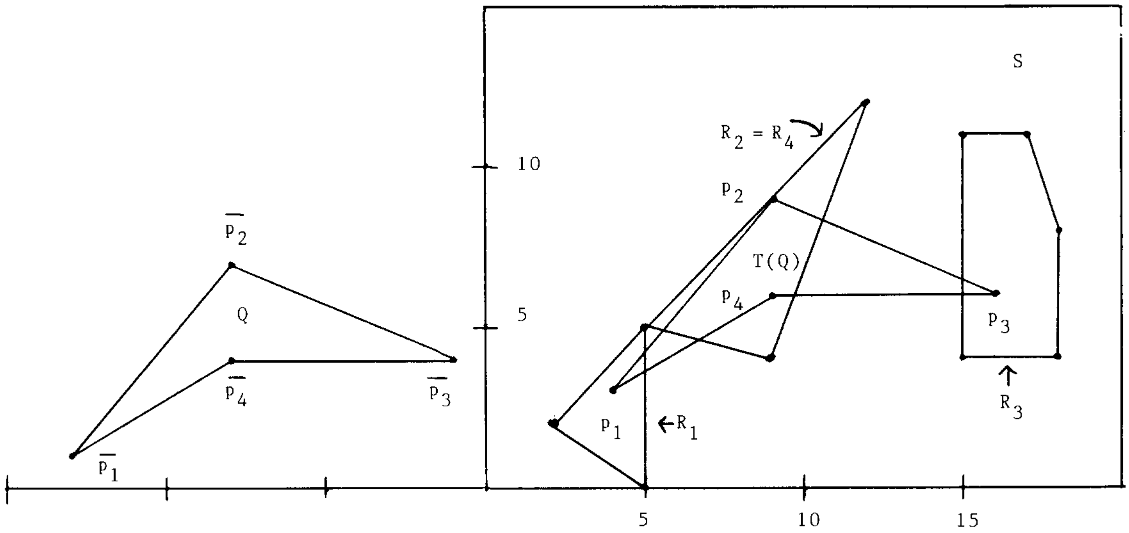

Figure 1a.

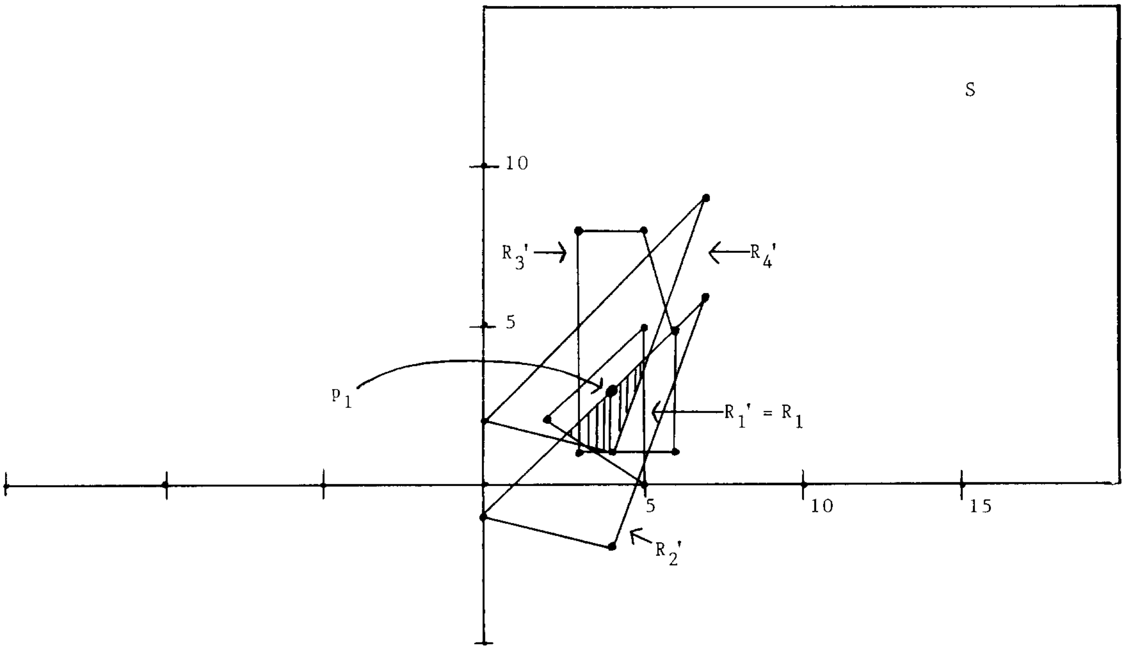

Figure 1b. The feasible region R is shaded. Note that the relative placement of p_i in R_i is the same as the placement of p_1 in R_i'.

3. The geometric question

The approach is to translate not the query polyhedron Q but the object polyhedra R_i. For each i, let T_i be that unique translation in $\mathbb{R}^r$ with $T_i(\bar{p}_i)=\bar{p}_1$. If we consider the points in $\mathbb{R}^r$ geometrically as vectors, then T_i is defined by $T_i(p)=p-c_i$, where $c_i=\bar{p}_i-\bar{p}_1$. Let $R_i'=T_i(R_i)$. Clearly, $R_1'=R_1$. Suppose that T is any translation of Q and let $p_i=T(\bar{p}_i)$ for each i. Then, for each i, $p_i-p_1=\bar{p}_i-\bar{p}_1=c_i$. Thus, for each i, $p_i=p_1+c_i$, and $p_i \in R_i'$ iff $p_1 \in R_i'$. Thus, there is a translation T of Q such that $T(\bar{p}_i)=p_i \in R_i$ for each i iff there is a translation T such that $T(\bar{p}_1)=p_1 \in R=R_1' \cap R_2' \cap \cdots \cap R_n'$. But this is possible iff R is nonempty. See Figure 1.

The original problem has been reduced to one of determining whether a certain convex polyhedron is nonempty. Suppose that each object polyhedron R_i, $1 \leq i \leq n$, is given as the intersection of closed halfspaces. Let $R_i=H_{i1} \cap H_{i2} \cap \cdots \cap H_{im_i}$, where each closed halfspace H_{ij} consists of those points $p \in \mathbb{R}^r$ which satisfy the corresponding linear inequality L_{ij} of the form $A_{ij}p^{\mathrm{T}}+b_{ij} \geq 0$. Here A_{ij} is a $1 \times r$ constant matrix, and b_{ij} is a scalar. This representation of the H_{ij} is a convenient uniform representation applicable to location spaces of dimension $r \geq 2$.

The object polyhedron R_i is thus the set of solutions to the system S_i given as the conjunction $S_i=L_{i1} \wedge L_{i2} \wedge \cdots \wedge L_{im_i}$. We note that $A_{ij}p_i^{\mathrm{T}}+b_{ij} \geq 0$ iff $A_{ij}(p_1+c_i)^{\mathrm{T}}+b_{ij} \geq 0$ and that this occurs iff $A_{ij}p_1^{\mathrm{T}}+(A_{ij}c_i^{\mathrm{T}}+b_{ij}) \geq 0$. Let us denote the linear inequality $A_{ij}p^{\mathrm{T}}+(A_{ij}c_i^{\mathrm{T}}+b_{ij}) \geq 0$ by L_{ij}'. Geometrically, the inequality L_{ij}' defines the halfspace $H_{ij}'=T_i(H_{ij})$ and the above equivalences express algebraically the fact that $p_i \in H_{ij}$ iff $p_1 \in H_{ij}'$. We note that the inequality L_{ij}' can be produced from L_{ij} in constant time. (We have assumed that the query polyhedron Q is predetermined, so the number of its vertices is fixed, and the vectors c_i can be computed in constant time.)

Let the system S_i' be given by the conjunction $S_i'=L_{i1}' \wedge L_{i2}' \wedge \cdots \wedge L_{im_i}'$. Geometrically, the system S_i' defines the polyhedron $R_i'=T_i(R_i)$. The system S_i' can thus be produced from the system S_i in time $\mathrm{O}(m_i)$ by an alteration made in each inequality separately.

Let S denote the system of inequalities formed by the conjunction $S=S_1' \wedge S_2' \wedge \cdots \wedge S_n'$. Then it follows directly from the discussion above that the system S (which defines the polyhedron R) can be produced from the systems defining the R_i in time $\mathrm{O}(m)$, where $m=m_1+m_2+\cdots+m_n$. See the Appendix.

The system S defines the feasible region R of a linear programming problem. The dimension is r (the dimension of the location space) and may be considered fixed. The method of Megiddo (1984) will find a point p in the feasible region (or characterize the region as empty) in time $\mathrm{O}(m)$. The location problem can thus be solved in time $\mathrm{O}(m)$. If R is nonempty, the required translation T is the unique translation with $T(\bar{p}_1)=p$. We note that the classical simplex method will (at worst) require time $\mathrm{O}(m^r)$ to solve the location problem.

Each possible translation T (each translation T of Q with $T(\bar{p}_i) \in R_i$ for each i) defines a possible location for the object O in the scene. It may be of interest to find all possible locations for O. That is, it may be of interest to actually characterize the feasible region R (not merely to determine whether it is nonempty). We note that Preparata and Muller (1979) have described an algorithm for finding the intersection of m halfspaces in $\mathbb{R}^3$ in time $\mathrm{O}(m \log m)$. The method is also applicable in $\mathbb{R}^2$.

A particularly simple case of the location problem occurs when each object polyhedron is a hyper-rectangle with sides parallel to the coordinate hyperplanes. In this case, each linear inequality L_{ij} defining the object polyhedron R_i is of the form $A_{ij}p^{\mathrm{T}}+b_{ij} \geq 0$, where the $1 \times r$ matrix A_{ij} has only one nonzero entry. Since Q is moved by translation, the corresponding polyhedron R_i' is also a hyper-rectangle with sides parallel to the coordinate hyperplanes. Each linear inequality L_{ij}' defining R_i' is, we recall, of the form $A_{ij}p^{\mathrm{T}}+(A_{ij}c_i^{\mathrm{T}}+b_{ij}) \geq 0$ and involves the same matrix A_{ij}. That is, each inequality in the final system S which defines the region R is an inequality involving one coordinate only. Since there are m such inequalities, the allowable range for each coordinate can thus be found (and the final region R thus characterized) by one scan of the system S and so in time $\mathrm{O}(m)$.

Such hyper-rectangles may be useful as containing regions in the case of generalized location conditions (Section 6) where several candidate locations are to be tried.

4. Optimal placement and general motion

If an object O appears in a scene, we may ask for the optimal position at which it appears. One reasonable meaning for the term 'optimal position' is that position which maximizes or minimizes some linear function of the coordinates in the location space. For instance, suppose that we are interested in an object which appears at a certain time and place in a two-dimensional scene, then remains. We might wish to find the object when it first appears. That is, we might wish to find the position of the object which minimizes the linear function $t = 0x + 0y + 1t$.

In general, we say that a placement of the object O in S is optimal with respect to some linear function g of the coordinates of the location space iff the placement minimizes or maximizes g at the vertex $\bar{p}_1$ of Q. That is, the placement is optimal iff the corresponding translation T is optimal in the sense that the function g is maximized (or minimized) over the feasible region at $p_1 = T(\bar{p}_1)$.

The problem of finding an optimal placement is, thus, the classical linear programming problem. Again, Megiddo's method may be used to find an optimal placement in time $O(m)$. There is no mathematical reason for there to be a unique optimal placement of Q. However, the circumstances of the underlying physical problem may guarantee uniqueness.

We note that the motion of Q need not be restricted to translation. What is required in the discussion of Section 3 is merely that, as Q is moved by a function T, each coordinate of each vertex $p_i = T(\bar{p}_i)$, $2 \le i \le n$, remain some linear function of the coordinates of $p_1 = T(\bar{p}_1)$. That is, for each i there should exist an $r \times r$ matrix B_i and a $1 \times r$ matrix C_i such that $p_i^{\mathrm{T}} = B_i p_1^{\mathrm{T}} + C_i^{\mathrm{T}}$ (where $B_1 = I_r$, the $r \times r$ identity matrix, and $C_1 = 0$). Q is thus allowed to move so that each vertex p_i remains the image of p_1 under some fixed affine transformation (the transformation being the identity when $i = 1$). This allows for quite general deformations of Q. (For a brief discussion of affine transformations, see Preparata and Shamos (1985, pp. 21-24).) Motion of Q by translation corresponds to the simple case in which each $B_i = I_r$. In the more general case, the system S which defines R can still be produced (by a simple alteration in each inequality separately) from the systems defining the R_i and the location problem solved in time $O(m)$.

5. Nonconvex object polyhedra

As previously noted, there is no reason to suppose that the object polyhedra R_i will always be convex or even connected. Suppose that each object region R_i is the solution set of a generalized system S_i of (nonstrict) linear inequalities L_{ij}, $1 \le j \le m_i$. By a generalized system we mean any Boolean expression in the inequalities L_{ij} in which we are allowed to use the connectives $\wedge$ and $\vee$ ('and' and 'or'). Negation, however, is not allowed. The negation of a nonstrict inequality defines an open halfspace, and we have restricted our attention to closed object regions. A generalized system describes a closed subset of the location space, possible nonconvex, possibly empty. Our original systems of Section 3 involved only the connective $\wedge$ and described convex polyhedra. In particular, a generalized system can describe any R_i which is the union of convex polyhedra of the type discussed in Section 3.

Once again we ask whether it is possible to translate the query polyhedron Q so that $p_i = T(\bar{p}_i) \in R_i$ for each i. We may use the substitution method of Section 3 to transform each generalized system S_i into a corresponding system S_i' such that the coordinates of p_i will satisfy S_i for each i iff the coordinates of p_1 satisfy S_i'. Since each individual inequality L_{ij} is transformed into a corresponding inequality L_{ij}', each system S_i' has the same Boolean form in the L_{ij}' as the system S_i in the original inequalities L_{ij}. We now ask whether the generalized system $S = S_1' \wedge S_2' \wedge \cdots \wedge S_n'$ has a solution. If so, the matching is possible, and the object is found.

We can reduce this situation to that of Section 3 by writing S as a disjunction of (ordinary) systems of linear inequalities $T_1, T_2, \ldots, T_k$. There will be a feasible solution to the generalized system S iff there is a feasible solution to one of the systems

T_j. We may consider each T_j as in Section 3. A straightforward approach to the decomposition is to consider S as a single Boolean expression whose atoms are the individual linear inequalities comprising the systems $S'_1, S'_2, \ldots, S'_n$. The distributive law can then be applied to reduce S to disjunctive normal form. Geometrically, the reduction corresponds to writing R as a union of closed convex polygons. There is no reason to suppose that the disjunctive normal form thus produced is minimal or even irreduntant. There do, of course, exist techniques for minimizing Boolean expressions which might be applied if the situation warrents. In the case of a two dimensional location space, the problem of decomposing a closed polygon as a union of closed convex polygons is of interest in other contexts and has been studied extensively. See, for instance, Preparata (1983) and Toussaint (1985).

In the case of nonconvex object polyhedra, it is expected that the dominating time for the solution of the location problem will be the decomposition time. It is also expected that the original Boolean forms describing the object regions R_i, and thus the decomposition time, will be determined by the actual physical scene (the actual distribution of feature values and how that distribution is best, or most easily, characterized).

6. Generalized location conditions

In the original location problem, we considered that the object O had been found in S iff it was possible to translate Q into S so as to place each vertex p_i in the corresponding region R_i. However, due to the actual circumstances of the underlying physical problem, it might be reasonable to consider the object O as found under more general conditions. For instance, we might consider O as found iff Q could be translated so that $p_1 \in R_1 \wedge (p_2 \in R_2 \vee p_3 \in R_3) \wedge p_4 \in R_4$. Since what constitutes a 'reasonable' location condition is so closely tied to the underlying physical circumstances, the determination of the condition for a particular application should be left to the user. We consider the problem in general.

Let B be some Boolean expression in the statements '$p_i \in R_i$', $1 \leq i \leq n$. By a generalized placement of the query polyhedron Q corresponding to B, we mean a translation of Q into S so as to make B true. Our original placement of Q corresponded to the particular Boolean expression $p_1 \in R_1 \wedge p_2 \in R_2 \wedge \cdots \wedge p_n \in R_n$.

Suppose that the polyhedron R_i is described by the generalized system of linear inequalities S_i, $1 \leq i \leq n$. We may use the substitution method of Section 3 to form the corresponding generalized systems S'_i such that the coordinates of p_i satisfy S_i iff the coordinates of p_1 satisfy S'_i. Suppose that in the Boolean expression B we formally replace each statement '$p_i \in R_i$' by the system S'_i. Let us call the resulting generalized system S the system corresponding to B. There is a generalized placement of Q in the scene corresponding to B iff the generalized system S has a solution. We may now apply the approach of the previous section.

We note that 'mixed' conditions such as '$p_1 \in R_2$' need not be considered separately. Since object regions may coincide, the same physical object region may be labelled in as many ways as convenient. Moreover, although we have tacitly assumed that the vertices $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_n$ of Q are distinct, in fact the same actual vertex may be labelled in several ways, and the previous arguments remain unchanged. Thus 'mixed' conditions may be removed by appropriate labelling.

Suppose that in the Boolean expression B the statement '$p_i \in R_i$' occurs k_i times. Then the generalized system S can be produced from the systems S_i in time $O(k_1 m_1 + k_2 m_2 + \cdots + k_n m_n)$. It is expected that the dominating time will be the time to decompose S into a disjunction of ordinary systems, and this time will depend on the original form of S. The location problem can then be solved in time $O(k)$, where k is the total number of inequalities appearing in the ordinary systems comprising S.

7. History and related results

In an early paper Chazelle (1983) considered a problem in computational geometry and in fact solved a special case of the location problem (with rotation of the query polygon allowed) in two dimensions. Chazelle supposed that P and Q were simple polygons and determined whether P could be moved by translation and rotation so as to fit into Q. If Q is convex, Q contains P iff Q contains all the vertices of P. Chazelle thus solved exactly the location problem with query polygon P and object polygons R_i which all coincide with Q. For nonconvex Q he solved the harder problem of placing each vertex p_i into R_i so as to place P entirely in the union of the R_i (again for the case in which each R_i coincided with Q). Applications of the harder problem to computer vision would be of interest.

Baird (1985) considered a similar problem (in two dimensions only). He began with a model consisting of points $p_1, p_2, \ldots, p_n$ and, surrounding each point p_i, a noise polygon N_i. He then determined whether this model could be matched to an instance consisting of points $q_1, q_2, \ldots, q_n$ by a feasible matching (one to one correspondence) M. The allowable motion was defined by a registration function R operating on the points q_i which allowed translation, rotation, and scaling. M was said to be feasible iff $R(M(p_i)) \in N_i$ for each i. That M be feasible imposed constraints on the parameters of the registration function R, and Baird used linear programming (the ellipsoid method) to determine whether there were parameters satisfying the constraints and so whether a candidate matching M was feasible.

In our context the model points p_i correspond to the vertices of the query polygon and the regions $R^{-1}(N_i)$ correspond to the object polygons. However, the regions $R^{-1}(N_i)$ are not given with the problem. They are not determined until M is known to be feasible and the registration function R is found. The geometric shape of the regions is, however, determined by the model (independent of any information about the scene). Baird gives a history of his problem which is of interest (pp. 10–13). There is a related paper by Nyo and Suk (1986) extending Baird's work to subpattern matching.

Grosky (1986) and Wu (1986) have solved the location problem in two dimensions by geometric methods involving angles of approach. W. Grosky has brought to my attention the independent work of I. Natour (1984). He has solved the problem in two dimensions with a time analysis based on work by Shamos (1975). The time complexity is $O(n \log N)$ if all object polygons are convex and $O(n^2 \log N)$ if at least one object polygon is nonconvex (where N is the number of object polygons and n is the total number of vertices). This is the time to characterize the set of possible locations for Q.

8. Conclusions

We began with the physical problem of locating a given object O in a scene S. We then showed how the physical problem could be reduced to a problem in computational geometry and, finally, to a linear programming problem. Under certain restrictions (namely that all object polyhedra R_i be convex) the problem can then be solved in time $O(m)$, where m is the total number of hyperplanes defining the R_i. The method of solution is applicable to location spaces of dimension $r \geq 2$.

The above is essentially a preliminary theoretical study. The next step is to find significant practical applications which exploit the full generality of the method, especially in higher dimensional location spaces.

Acknowledgments

I would like to thank the referee for suggesting several improvements in the paper. I would also like to thank W. Grosky for the geometric question itself and for his many helpful comments. Finally, I would like to thank I. Sethi and N.K. Tsao for their suggestions.

The basic results were obtained during the 1985–86 academic year when I was a Visiting Assistant Professor of Computer Science at Wayne State University.

Appendix

In Figure 1, the regions R_i are given by the following systems of inequalities:

$$\begin{array}{llrcl}
R_1: & 4x + 5y - & 20 & \geq & 0, \\
 & -x \quad + & 5 & \geq & 0, \\
 & x - \quad y & & \geq & 0; \\
R_2 = R_4: & x - \quad y & & \geq & 0, \\
 & 7x + 20y - & 140 & \geq & 0, \\
 & -3x \quad y + & 24 & \geq & 0; \\
R_3: & x \quad - & 15 & \geq & 0, \\
 & -x \quad + & 18 & \geq & 0, \\
 & y - & 4 & \geq & 0, \\
 & - \quad y + & 11 & \geq & 0, \\
 & -4x - \quad y + & 80 & \geq & 0.
\end{array}$$

Thus, in matrix form, the determining conditions for the points $p_i = (x_i, y_i)$ are as follows:

$$R_1: \begin{bmatrix} 4 & 5 \\ -1 & 0 \\ 1 & -1 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} -20 \\ 5 \\ 0 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix};$$

$$R_2: \begin{bmatrix} 1 & -1 \\ 7 & 20 \\ -3 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -140 \\ 24 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad c_2 = (5,6);$$

$$R_3: \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ -4 & -1 \end{bmatrix} \bullet \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} + \begin{bmatrix} -15 \\ 18 \\ -4 \\ 11 \\ 80 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad c_3 = (12,3);$$

$$R_4: \begin{bmatrix} 1 & -1 \\ 7 & 20 \\ -3 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_4 \\ y_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -140 \\ 24 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad c_4 = (5,3).$$

Finally, using the fact that $p_i = p_1 + c_i$, $2 \leq i \leq 4$, the determining conditions for the point $p_1 = (x_1, y_1)$ are as follows. These conditions define the feasible region R.

$$R_1 = R_1': \begin{bmatrix} 4 & 5 \\ -1 & 0 \\ 1 & -1 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} -20 \\ 5 \\ 0 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix};$$

$$R_2': \begin{bmatrix} 1 & -1 \\ 7 & 20 \\ -3 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} -1 \\ 15 \\ 15 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix};$$

$$R_3': \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ -4 & -1 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} -3 \\ 6 \\ -1 \\ 8 \\ 29 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix};$$

$$R_4': \begin{bmatrix} 1 & -1 \\ 7 & 20 \\ -3 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} 2 \\ -45 \\ 12 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

The solution $p_1 = (x_1, y_1) = (4,3)$ is the solution indicated in the sketch in Figure 1.

References

Baird, H.S. (1985). *Model Based Image Matching Using Location.* MIT Press, Cambridge, MA.

Chazelle, B. (1983). The polygon containment problem. In: F.P. Preparata, Ed., *Advances in Computing Research Vol. 1: Computational Geometry.* JAI Press, Greenwich, CT.

Grosky, W. (1986). Personal communication.

Megiddo, N. (1984). Linear programming in linear time when the dimension is fixed. *J. ACM* 31, 114-127.

Natour, I. A translation of a polygon with a constrain (sic). Unpublished manuscript.

Nyo, H.L. and M. Suk (1986). A polynomial time algorithm for subpattern matching. *Proc. IEEE* 74, 375-377.

Preparata, F.P., Ed. (1983). *Advances in Computing Research Vol. 1: Computational Geometry.* JAI Press, Greenwich, CT.

Preparata, F.P. and D.E. Muller (1979). Finding the intersection of n halfspaces in time $O(n \log n)$. *Theoret. Comput. Sci.* 8, 45-55.

Shamos, M.I. (1975). Geometric complexity. *Proceedings of the 17th Annual ACM Symposium on the Theory of Computing,* 224-233.

Toussaint, G.T., Ed. (1985). *Computational Geometry.* North-Holland, New York.

Wu, S. (1986). A dynamic geometric object intersection problem. Wayne State University Department of Computer Science (masters thesis), Detroit, MI.