DTIC FILE COPY

AD-A202 284

(4)

Systems
Optimization
Laboratory

Track Initialization in the Multiple-Object Tracking Problem

by

Karel Zikan

TECHNICAL REPORT SOL 88-18

November 1988

Department of Operations Research
Stanford University
Stanford, CA 94305

(4)

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-4022

Track Initialization in the Multiple-Object Tracking Problem

by
Karel Zikan

TECHNICAL REPORT SOL 88-18

November 1988

Research and reproduction of this report were partially supported by the National Science Foundation Grant DMS-8800589; U.S. Department of Energy Contract DE-FG03-87ER25028, and Office of Naval Research Contract N00014-85-K-0343.

TRACK INITIALIZATION IN THE MULTIPLE-OBJECT TRACKING PROBLEM

Karel ZIKAN*

Department of Operations Research
STANFORD UNIVERSITY
Stanford, CA 94305

Abstract

The multiple-object tracking problem involves extraction of the trajectories of n moving points from (three) successive motion picture frames. In the track initialization part of the problem no previous history of track evolution is given.

A definition of a "three-point metric" functional (analogous to the classical definition of distance) is put forward. For the best estimate of the trajectories, we partition the points from the frames into n triplets (based on the three successive frames) so that the average three-point "distance" is minimized. The physical intuition behind this approach is discussed and several equivalent mathematical programming formulations are given. A practical method proposed for solving of the problem is based on a Lagrangean relaxation technique, and, to a lesser degree, on the "pruning" of the tree of "subpartitions". On the basis of empirical evidence and experience from related work, we conjecture that, on average, $O(n^3)$ arithmetic operations are needed to obtain a solution. The problem of missing and spurious points in the images is also briefly discussed. A short summary and examples from simulated data experiments are given.

* Karel Zikan is also with the Hughes Artificial Intelligence Center, Hughes Aircraft Company.

0. Introduction

In this paper we address the *multiple object tracking problem.* We are given three consecutive frames (one can think of a movie) of the same scene. The frames are taken in regular fraction-of-a-second intervals and contain n moving points. The *track initialization problem* is to estimate the n most probable trajectories that may give rise to the given set of frames. It is called track initialization problem because no "history" of the previous tracks' evolution is given. The track initialization problem is a harder problem than the also important *track extension* problem, where in loose terms we need to extend n existing trajectories so that they would incorporate the set points of the latest frame. In general, both of these subproblems of the *multiple-object tracking problem* arise naturally in the aerospace and defense industries. For instance, see ARNOLD, BAR-SHALOM, and MUCCI [2].

The "three-point metric" approach to the track initialization problem came up as an application of *algebraic metric theory*, ZIKAN [14]. Chapter 12 of [14] contains the information presented here. The rest of [14] contains supporting material and applications of the theory to related computer vision problems (e.g., *detection of missing and spurious points*). However, the research presented here can be understood without studying it in the broader context of the algebraic metric theory.

In Section 1 we introduce the notion of "distance" between three points simultaneously. We provide a definition and some examples.

In Section 2 we begin the discussion of the track initialization problem. We provide the motivation, discuss the goals, and identify the difficulties. Here also the track initialization and track extension problems are contrasted. The principle of momentum preservation provides the key physical insight and motivates the model. We choose one of the three-point "metrics" to serve as a "penalty function" for the estimated trajectories. The n (disjoint) trajectories that together give the least total penalty become our estimate.

In Section 3 we propose a mathematical programming model, and give an algorithm to solve it. The proposed solution method has two stages. During the first stage, a Lagrangean relaxation technique is used to obtain a stationary point. Although the point need not be the global optimum, it usually will be. To insure global optimality as well as to find good alternative solutions, a branch and bound tree pruning algorithm with the stationary point as the initial incumbent is proposed. The overall average case "running time" seems to be about $O(n^3)$.

The computational complexity issues are discussed in Section 4. The discussion focuses mostly on the empirical average case running time requirements.

Section 5 contains a description of a simulated track initialization experiment. We briefly discuss the problem of missing and spurious points. The method developed in [14, Chapter 10], is used here as a subroutine to "clear" the images of spurious points. Examples and figures of the track initialization problem in various stages of computation are presented in the remainder of this section.

1. Three-Point "Distances"

Distances are measures of *closeness* or *similarity*. Given a space X, a metric d can be viewed as a functional that maps pairs of points in X into real numbers, $d : X \times X \to R$, and that reflects the "degree of association" of each pair of points. This interpretation of metrics leads inevitably to the idea of "distances" *between three* (or *more*) points. We want to define a *three-point "metric"* $\mathcal{D}$ as a functional that maps each three points into "reals", $D : X^3 \to R$, and that captures the *degree of association* for all triplets of points.

Definition: A functional $\mathcal{D}$ is a *three-point* (3-p) *metric* on the space X if it satisfies the following properties for all ordered triplets of points $x, y, z \in X$:

P1: $\mathcal{D}(x, y, z) \geq 0$,

P2: $\mathcal{D}(x, y, z) = \mathcal{D}(z, y, x)$,

P3: $\mathcal{D}(x, x, x) = 0$,

P4: $\inf_{w \in X} \mathcal{D}(x, w, y) + \inf_{w \in X} \mathcal{D}(y, w, z) \geq \inf_{w \in X} \mathcal{D}(x, w, z)$.

These axioms are aesthetically pleasing for the way they correspond to the normal (two-point) axioms. They are also useful as we will see later in the chapter. The definition generalizes to more than three points in a transparent, nested, manner. Here are some examples of 3-p metrics.

Example 1: Perimeter 3-p metric. Given a normed space $\{X, \|\cdot\|\}$, we define $\mathcal{D}(x, y, z) = \|x - y\| + \|y - z\| + \|x - z\|$, the perimeter of the triangle $\Delta(x, y, z)$. Note that **P1-P3** are clearly satisfied. If y is a convex combination of x and z then $\inf_{y \in X} \mathcal{D}(x, y, z) = 2\|x - z\|$ is achieved. Thus the triangle inequality, **P4**, holds too. Note that the axioms are also satisfied if the $\|x - z\|$ term is weighted. If, instead of a normed space, we are given an arbitrary metric space $\{X, d\}$, the triangle perimeter still defines a 3-p metric. The infimum $\inf_{y \in X} \mathcal{D}(x, y, z) = 2d(x, z)$ is still achieved for $y = x$.

Example 2: Centroidal 3-p metric. *Consider the euclidean space* E^n. Let c be the centroid of $x, y,$ and z and define $\mathcal{D}(x, y, z) = \|x - c\| + \|y - c\| + \|x - c\|$, the sum of the distances to the center of the triangle $\Delta(x, y, z)$. Again, **P1-P3** are clearly satisfied. Note, that $\inf_{y \in R^n} \|c - y\| = 0$ is achieved when $y = c$, and $\inf_{c \in R^n} \|x - c\| + \|c - z\| = \|x - z\|$ is achieved when $c = \frac{x+z}{2}$. Therefore, if $y = c = \frac{x+z}{2}$, the $\inf_{y \in R^n} \mathcal{D}(x, y, z) = \|x - z\|$ is achieved. We can now see that $\mathcal{D}$ satisfies **P4**.

Example 3: Inertial 3-p metric. For the euclidean space E^n and scalar $\alpha > 0$, we define

$$\mathcal{D}_\alpha(x, y, z) = \|x - y\| + \|z - y\| + \alpha\|\frac{x + z}{2} - y\|. \qquad (1-1)$$

Figure 1.1 shows the line segments lengths of which are the components of the functional $\mathcal{D}(x, y, z)$. The functional is a 3-p metric, as it is trivial to verify the axioms **P1-P3**, and the "triangle inequality", **P4**, follow easily after noticing that $\inf_{y \in R^n} \mathcal{D}(x, y, z) = \|x - z\|$ is achieved when $y = \frac{x+z}{2}$. This metric is used in our approach to the track initialization problem.

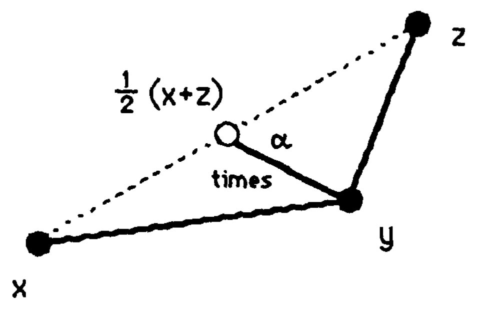


Figure 1.1 *Components of the inertial 3-p metric*

2. Track Initialization Problem

The essence of the *track initialization problem* is to "match" triplets of points — one from each image frame — so that the collection of trajectories producing these triplets best satisfies our notion of reasonableness.

From the practical standpoint, track initialization is a difficult problem only because of technological limitations. If we could take the frames in a fast enough sequence, effectively making a "movie", then the extraction of the trajectories would be simple. Unfortunately, for some applications, the camera speeds are inadequate and the distance between trajectories may be less than the distance traveled by points between the frames. Some trajectories may also cross in the time before the second frame is taken.

The principal idea of the method proposed to solve the track initialization problem is simple. Physical objects do not move with arbitrary speeds, and they obey the inertial laws. The velocity should change only continuously and gradually. Good trajectories must satisfy these requirements. If p, q, and r are "successive" points belonging to the same trajectory, it is natural to require that (in the terms of the usual euclidean distance):

i. the points be close to each other;
ii. the middle point q be close to $q' = \frac{p+r}{2}$, the point halfway between p and r. (2 − 1)

We build our model around these requirements.

Recall the 3-p metric of Example 1.3. With each triplet of points we associate as a penalty the function (1-1) with the default value of α specified as $\alpha = 4$. We need to find a set of n (disjoint) triplets of points which minimizes the total penalty. There are exactly $(n!)^2$ feasible sets available. Therefore an explicit enumeration of all possibilities is out of the question for all but the smallest problems. (With ten objects there are more than 10^{13} feasible sets of triplets.)

The rationale for using the 3-p metric (1-1) as the penalty comes from the assumption that the frames were taken at the (technologically) fastest possible rate. Large displacements as well as large changes in velocity are thus unacceptable. The functional $\mathcal{D}_\alpha$ "penalizes" high speed and acceleration in trajectories.

Let us (generically) refer to the points of the first, second, and third frame by p^i, q^j, and r^k respectively. We suppress the parameter α and use the shorthand notation, $\mathcal{D}_{ijk} = \mathcal{D}_\alpha(p^i, q^j, r^k)$. The first two terms of $\mathcal{D}_{ijk}$, the values $||p^i - q^j||$ and $||r^k - q^j||$, represent the penalty for displacement. The last term, $\alpha||\frac{p^i + r^k}{2} - q^j||$ is the penalty for inertia change.

The value of α allows a relative weighting of the respective parts of the penalty function. The choice of α is slightly arbitrary. In fact, the weight parameters provide a user with a simple means of control. By changing α, we alter the method's degree of preference for linear and monotone (no acceleration) motion. The default weight, $\alpha = 4$, is motivated by the geometry of the problem.† In (stylized) experiments on trajectories generated by a set of parametric curves (Section 5), only the default weight was used. The correct trajectories were selected in all test problem instances.

† Consider the *collinear* triplet of points, p, q, and r, produced by an accelerating point. Since we assume that the frames are taken in regular 1-second intervals (let $t = 1$), $v = \frac{1}{2}(||q - p|| + ||r - q||)$ is the estimate of the average velocity, and $a = ||r - q|| - ||q - p||$ is the estimate of the acceleration. Because the points are collinear a simple computation verifies that $\mathcal{D}_4(p, q, r) = 2(v + ta)$.

3. Formulations and Computational Methods

Integer linear programming formulation. The integer linear program,

$$\begin{aligned} \min_{x} & \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \mathcal{D}_{ijk} x_{ijk} \\ \text{subject to} & \sum_{i=1}^{n} x_{ijk} = 1, \text{ for all pairs } jk, \\ & \sum_{j=1}^{n} x_{ijk} = 1, \text{ for all pairs } ik, \\ & \sum_{k=1}^{n} x_{ijk} = 1, \text{ for all pairs } ij, \\ \text{and } & x_{ijk} \in \{0,1\}, \end{aligned} \tag{3-1}$$

is the mathematical description of the formulation described above. It is a generalization of the assignment problem, PAPADIMITRIOU and STEIGLITZ [13]. The variable $x_{ijk} = 1$ if and only if the triplet of points p^i, q^j, and r^k is grouped together. Because the constraint matrix of (3-1) is not *unimodular* we can not relax the integrality constraint. The optimal solution set to the linear programming relaxation need not contain any integral vertices, GARFINKEL and NEMHAUSER [8]. This is an unfortunate computational complication in comparison to the assignment problem.

Bilinear programming formulation. We can avoid the integrality constraints if we formulate the track initialization problem as an equivalent bilinear programming, KONNO [11], problem:

$$\begin{aligned} \min_{x,y} & \sum_{ijk} \mathcal{D}_{ijk} \cdot x_{ij} \cdot y_{jk} \\ \text{subject to} & \sum_{j=1}^{n} x_{ij} = 1, \text{ for all } i, \\ & \sum_{i=1}^{n} x_{ij} = 1, \text{ for all } j, \\ & x \geq 0 \\ \text{and to} & \sum_{j=1}^{n} y_{jk} = 1, \text{ for all } k, \\ & \sum_{k=1}^{n} y_{jk} = 1, \text{ for all } j, \\ & y \geq 0, \end{aligned} \tag{3-2}$$

where the indices range over the integers $\{1, 2, \ldots, n\}$. Formulations (3-1) and (3-2) are equivalent in the sense that from a solution of (3-1), we can construct a *cornerpoint* solution of (3-2) and vice versa. Furthermore, if we are given a solution of (3-2) which is not a cornerpoint of the feasible region, then it is a simple matter *($O(n^3)$ worst case complexity) to obtain a cornerpoint* solution of (3-2) as well. These aspects of multilinear programming are discussed in Section 8.3 of ZIKAN [14].

Trilinear programming formulation. There is a certain lack of symmetry inherent in the formulation (3-2). This can be rectified by formulating the problem as (yet another) equivalent mathematical programming problem. It is the following *trilinear* program

$$\begin{aligned}
&\min_{x,y,z} \sum_{s=1}^{n} \sum_{ijk} \mathcal{D}_{ijk} \cdot x_{si} \cdot y_{sj} \cdot z_{sk} \\
&\text{subject to } \sum_{s=1}^{n} x_{si} = 1, \text{ for all } i, \ \sum_{i=1}^{n} x_{si} = 1, \text{ for all } s, \ x \geq 0, \\
&\qquad \sum_{s=1}^{n} y_{sj} = 1, \text{ for all } j, \ \sum_{j=1}^{n} y_{sj} = 1, \text{ for all } s, \ y \geq 0, \\
&\qquad \sum_{s=1}^{n} z_{sk} = 1, \text{ for all } k, \ \sum_{k=1}^{n} z_{sk} = 1, \text{ for all } s, \ z \geq 0,
\end{aligned} \tag{3-3}$$

where, again, the indices range over the integers $\{1, 2, \ldots, n\}$.

Of all these formulations of the track initialization problem, the trilinear mathematical programming problem (3-3) makes the best computational use of the special structure. The feasible region is a *direct sum* of the three orthogonal polyhedra. The only coupling of the variables x, y, and z is via the inertial term of the objective function. It prohibits a complete decomposition into three independent assignment problems.

Lagrangean cyclic block descent method. Note that if we fix two of the variables, e.g., y and z, then (3-3) reduces to an assignment problem. This subproblem can be solved by any of the standard methods. The availability of these easily solvable subproblems suggests an algorithm built on the Lagrangean relaxation technique. In this case, the Lagrangean block descent algorithm involves cyclically fixing two variables at a time and optimizing over the third variable. At each stage we obtain a better (cornerpoint) solution as the objective function decreases. We continue iterations of the algorithm until there is no improvement in the objective function for an entire relaxation cycle. At this stage, we have obtained a *stationary solution*, x^*, y^*, z^*. As there is only finite number of cornerpoint solutions the Lagrangean relaxation algorithm will terminate finitely.

The number of major cycles through the procedure varies with the quality of the initial starting point. Empirical evidence suggests that the average case growth might be as low as logarithmic in the number of points, n. A linear bound on the average case growth seems to be a conservative estimate. More study in this direction needs to be done.

It can not be guaranteed that the stationary solution will be a global optimum. However, in the tests reported here, the global minimum was found by the Lagrangean algorithm every time.

Tree-pruning method of implicit enumeration. After the initial stationary solution is found, we can effectively use a tree-pruning approach to verify the overall optimality of the solution. The method is fashioned after a common tree-pruning strategy which can be found in BAIRD [3], ZIKAN and SILBERBERG [15], and in many other papers cited therein. The order of the points $p^1, p^2, \ldots, p^n$ is fixed. There are n^2 pairs of points (q^j, r^k). Any of these pairs can be matched to the point p^1. Those matches that produce too large a penalty can be discarded from consideration. No successful completion of the triplet selection exists. (An implicit enumeration of a large set of feasible solutions has been accomplished.) We now extend one of the remaining matching candidates to another level. Without loss of generality assume that the partial tree consists of the triplet (p^1, q^1, r^1). At this stage candidates from the $(n-1)^2$ remaining pairs are matched to the point p^2. For each candidate the penalty is added to the penalty associated with the first triplet. If the penalty exceeds the threshold, the branch can be pruned again. We proceed on in the depth-first manner.

The stationary matching obtained by the relaxation method is used as the initial incumbent in the tree-pruning branch and bound method. Examination of the search tree of partial solutions produces a guaranteed optimal solution. (Chances are that the Lagrangean solution will be returned as optimal.) The analysis of Baird [3] for the image registration problem suggests that the pruning of the tree can be done in $O(n^3)$ average "time". The good starting incumbent and the natural "steepness" of the penalty function (that is, even with a single mismatched triplet the associated penalty is typically large) together facilitate a strong fathoming criterion. This conjecture deserves empirical as well as analytical study.

In addition to its capability to return the optimal answer, the value of the tree-pruning search lies in the fact that all plausible matching alternatives can be discovered. During the search through the tree, the fathoming criterion can be adjusted so that all assignments with penalty within ξ% of the optimal value are returned. With equal ease we can return all solutions differing from the optimal one by a certain amount, we can return the k best solutions, and so on.

4. Computational Complexity of the Methods

The size of the problem is n, the number of points in a frame. Evaluating the penalty for each triplet of points requires $O(n^3)$ operations. Therefore an algorithm which computes the answer to (3-3) in approximately $O(n^3)$ time is very efficient.

The computational procedure has two parts: the Lagrangean section (producing the initial stationary point) and the branch and bound section of tree pruning. The complexity issue of each section can be further decomposed into smaller units.

Complexity of finding the stationary point. The Lagrangean part makes essential use of the linear assignment problem. It takes $O(n^2)$ time to formulate an assignment problem and $O(n^3)$ worst case time to solve it. The average time required to solve the assignment problems is somewhere between $O(n^2)$ and $O(n^3)$, KENNINGTON and HELGASON [10].

In order to estimate the average time to solve the entire Lagrangean problem, we need to consider the expected number of the assignment problems, and the expected time to solve the assignment problems using the "hot" start technique. At the moment we can only offer a conjecture. It appears that the overall average case complexity of finding the stationary point is $O(n^3)$ or better.

If the above conjecture is true, we get the nearly paradoxical result that the time to find a stationary solution may be in the average sense even faster than $O(n^3)$, the time to evaluate the penalty for all possible triplets. To explain this seemingly impossible result we need to look back at the equation (1-1). The definition says that that only the "inertial portion" of the penalty involves all three points. Therefore if the distance between a pair of points is quite large, it becomes possible to *implicitly* fathom all triplets containing the given pair of points. The inertial terms need not be evaluated. (To enumerate distances between all pairs of points we need only $O(n^2)$ arithmetic operations, of course.)

Tree-pruning method. The time required to descend a full tree branch is $O(n^3)$. There is no distinction between the average case and between the worst case complexity.

Although there are $(n!)^2$ leaves in the tree, experience from problems with similar flavor indicates that all but a small number of branches are pruned in their upper levels. A conjecture is that the pruning will also require $O(n^3)$ operations on the average.

5. Experiment Description

A Note on Missing and Spurious Points. All (computer vision) practitioners know that a method that falls apart when missing or spurious points are present in the problem is of academic importance only. In the track initialization problem, missing and spurious points are inherent. Even if there were no other technological reasons, we would still have the problem that objects may either enter the field of view or exit from it.

The problem of detecting (missing and) *spurious points* was treated in ZIKAN [14, Chapter 10]. In summary, the method developed there was based on a combination of ideas from linear programming geometry and from information theory. By "detection" we mean pointing out the most promising candidate to be the spurious point. In reality, the candidate may not be *the spurious* point, but it usually is. The crude, preliminary version implemented as a subroutine in the track initialization problem detected the correct spurious points in all but in one instance.

In our tests the detected spurious points were deleted from consideration for matching. Due to the tests' small problem sizes, the fraction of spurious points could become quite high. At one time 2 out of the 8 (25%) points were spurious. This case was solved successfully, too.

Simulated experiment. We can now look at simulations of the track initialization problem. A set of two-dimensional parametric curves was generated and each was given an initial starting point. The family included among others: *line, circle, cardioid, inverted parabola, figure "8", cycloid, inverted square root, logarithmic curve,* and a number of *concatenations of sines and cosines.* Some curves stayed on the screen all time, some entered, passed, and then exited from view.

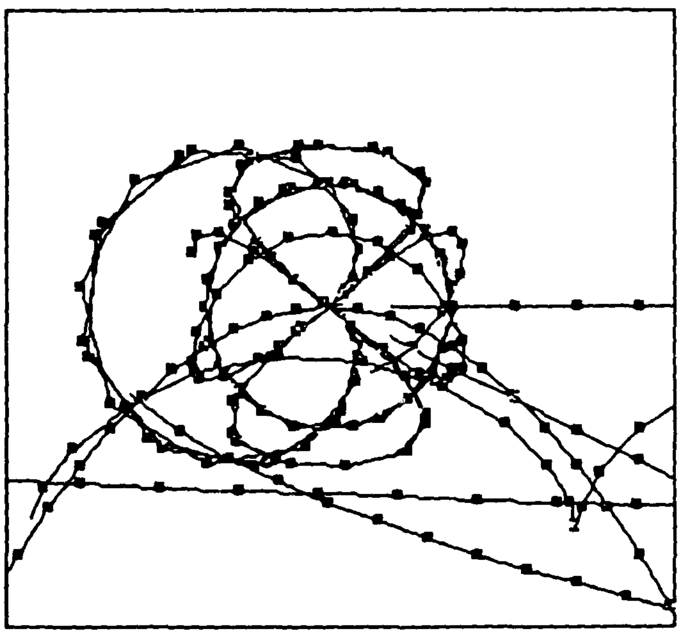


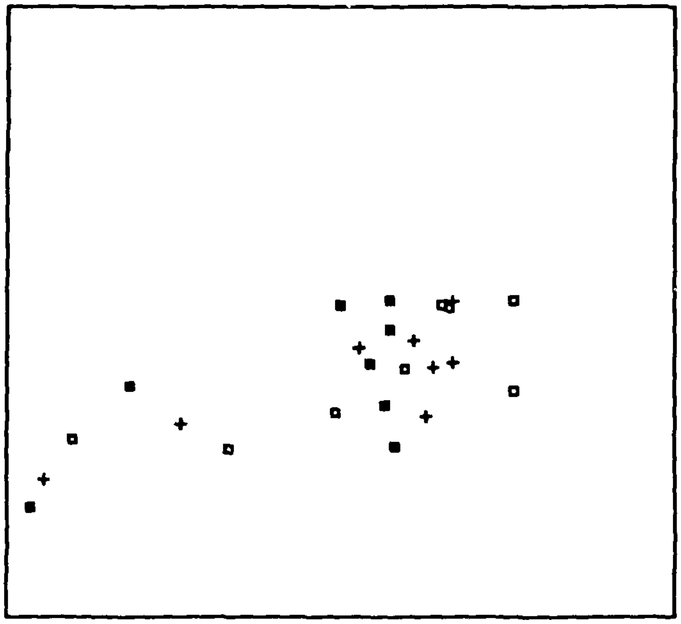
Figure 5.1 *Tracks of experiment's trajectories*

Figure 5.1 exhibits all relevant trajectories as they moved across the screen. On the average we had 8 curves in "sight" at any given time.†

Figure 5.2 captures the situation when no spurious point complicates the scene. All three consecutive frames are overlaid; the "oldest" points are marked by full squares, the "middle aged" points by crosses, and the "youngest" points are marked by squares with empty interiors. Figure 5.2.a) is the picture of the data submitted to the solving subroutine. Figure 5.1.b) exhibits the (correctly) estimated trajectories. A natural challenge for the reader is to estimate the solution with the help of the a) display only.

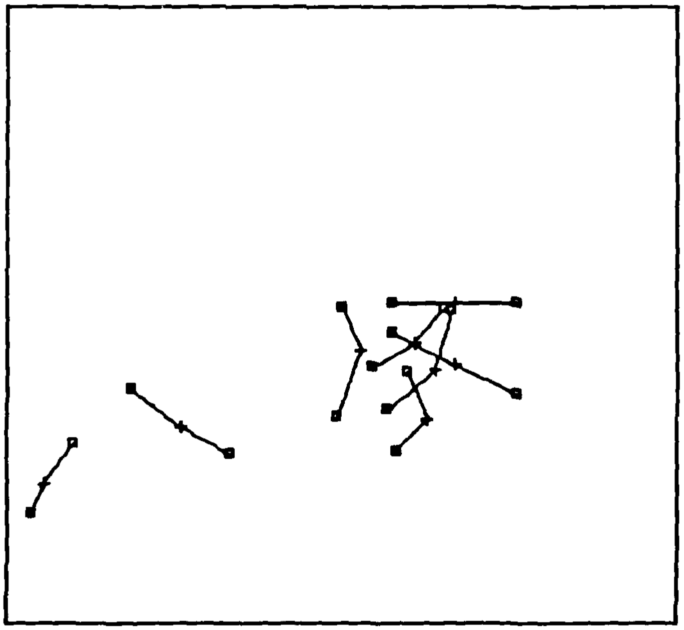
Figure 5.3 depicts the situation of the same set of trajectories a few frames "later" when two curves were concurrently leaving the scene. The Figure 5.4 comes from a still later time when a new curve entered the visual field. Note that the entering and leaving points are left unmatched. Otherwise the discussion from Figure 5.2 applies, too.

† The number may seem low to a person who has not tried to do the track initialization by hand.



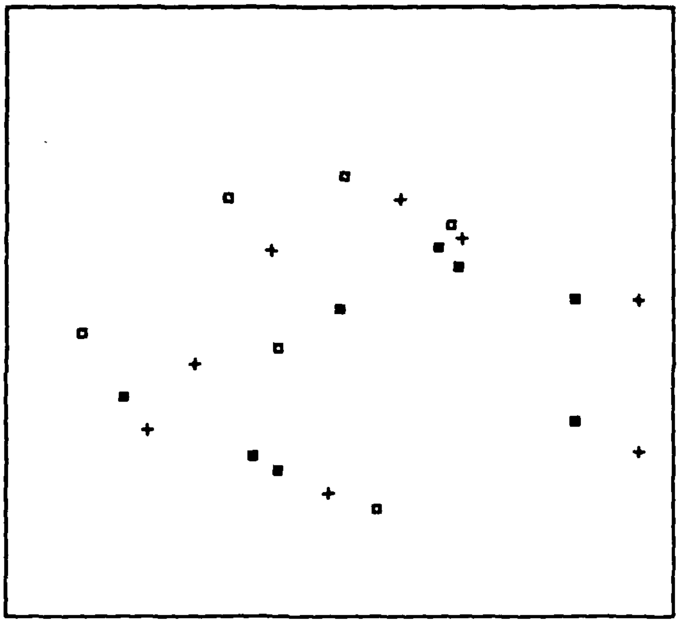
a)

■ frame 1, + frame 2, ▫ frame 3

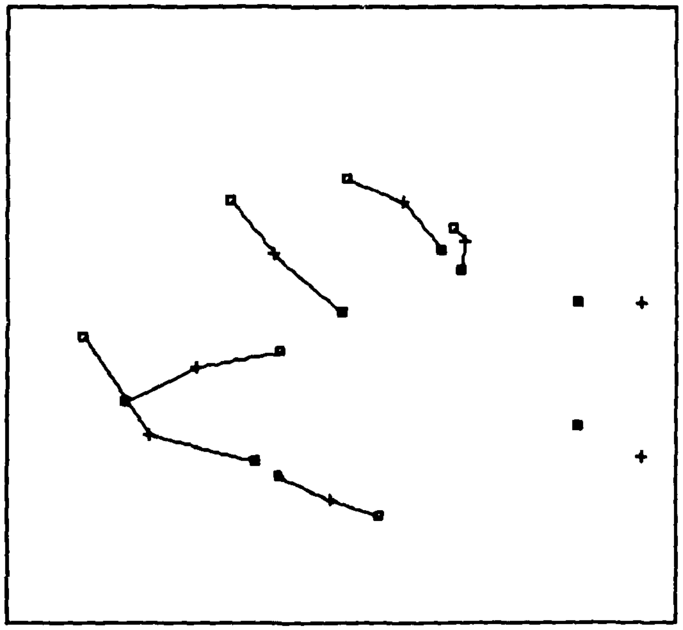


b)

Figure 5.2 *Example of track initialization problem*

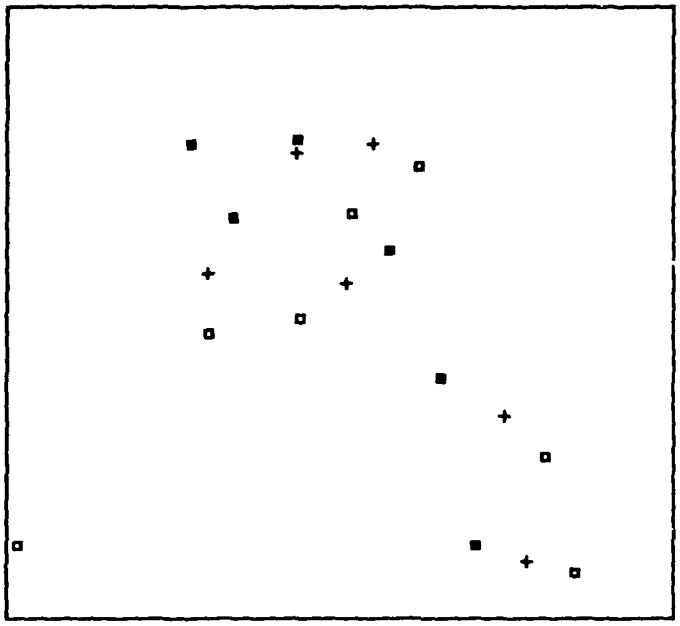


a)

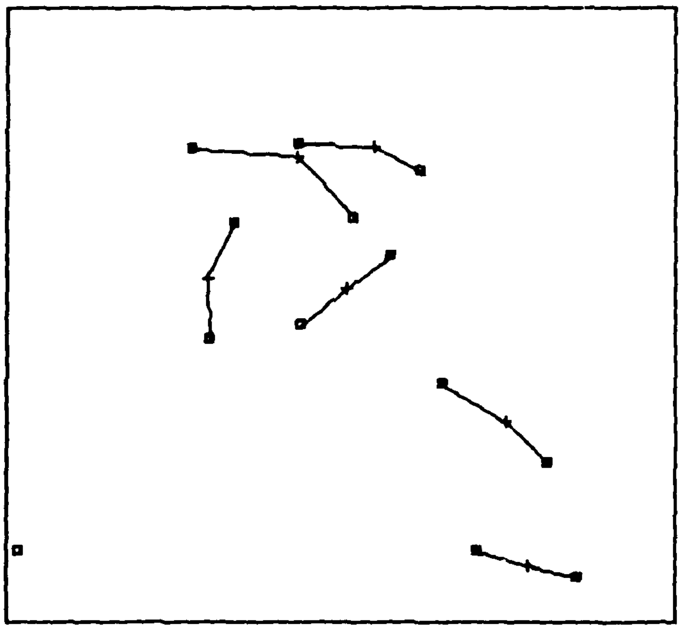


b)

Figure 5.3 *Example of track initialization problem with two tracks simultaneously "leaving"*



a)



b)

Figure 5.4 *Example of track initialization problem with one track "entering"*

6. References

[1] M. Altman, Bilinear programming, *Bulletin de l' Académie Polonaise des Sciences* **16** **(9)**, 741-746, 1968.

[2] J. Arnold, Y. Bar-Shalom, and R. Mucci, Track segment association with a distributed field of sensors, *Proc. of the Amer. Control Conference*, 1984.

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

[4] D.P. Bertsekas and P. Tseng, Relaxation methods for minimum cost ordinary and generalized network flow problems, *Operations Research* **36**, 1988, 93–115.

[5] O.D. Faugeras and M. Herbert, 3-D Shapes from Range Data, *Techniques for 3-D Machine Perception; A. Rosenfeld (Editor)* **3**, Elsevier Science Publishers B.V. (North-Holland), 1986, pp. 13–51.

[6] O.D. Faugeras and K.E. Price, Semantic description of aerial images using stochastic labeling, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **3**, 1981, pp. 633–642.

[7] T.E. Flick and L.K. Jones, A combinatorial approach for classification of patterns with missing information and random orientation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**, 1986, pp. 482–490.

[8] R.S. Garfinkel and G.L. Nemhauser, *Integer Programming*, John Wiley & Sons, New York, 1972.

[9] R. Gallo and A. Ülkücü, Bilinear programming: an exact algorithm. *Mathematical Programming* **12**, 173-194, 1977.

[10] J. L. Kennington and R. V. Helgason, *Algorithms for Network Programming*, John Wiley & Sons, New York, 1980.

[11] H. Konno, Bilinear programming, Ph.D. Dissertation, Dept. of Operations Research, Stanford University, 1971.

[12] H. Konno, A cutting plane algorithm for solving bilinear programs, *Mathematical Programming* **11**, 15-27, 1976.

[13] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[14] K. Zikan, Algebraic metric theory and applications, Ph.D. Dissertation, Department of Operations Research, Stanford University, California, forthcoming.

[15] K. Zikan and T. M. Silberberg, The Frobenius Metric in Image Registration. Technical Report SOL 88-13, Department of Operations Research, Stanford University, California, 1988.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report SOL 88-18	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Track Initialization in the Multiple-Object Tracking Problem		5. TYPE OF REPORT & PERIOD COVERED **Technical Report**
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Karel Zikan		8. CONTRACT OR GRANT NUMBER(s) N00014-85-K-0343
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research - SOL Stanford University Stanford, CA 94305-4022		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1111MA
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research - Dept. of the Navy 800 N. Quincy Street Arlington, VA 22217		12. REPORT DATE November 1988
		13. NUMBER OF PAGES 15 Pages
		15. SECURITY CLASS. (of this report) **UNCLASSIFIED**
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Track Initialization, Multiple Object Tracking Problem; Inertial "Three Point Metric"; Multilinear Programming; Lagrangean Relaxation. (edc)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Please see back for abstract...

DD FORM 1 JAN 73 1473 A EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Track Initialization in the Multiple-Object Tracking Problem

by Karel Zikan

Technical Report SOL 88-18 Abstract

Abstract

The multiple-object tracking problem involves extraction of the trajectories of n moving points from (three) successive motion picture frames. In the track initialization part of the problem no previous history of track evolution is given.

A definition of a "three-point metric" functional (analogous to the classical definition of distance) is put forward. For the best estimate of the trajectories, we partition the points from the frames into n triplets (based on the three successive frames) so that the average three-point "distance" is minimized. The physical intuition behind this approach is discussed and several equivalent mathematical programming formulations are given. A practical method proposed for solving of the problem is based on a Lagrangean relaxation technique, and, to a lesser degree, on the "pruning" of the tree of "subpartitions". On the basis of empirical evidence and experience from related work, we conjecture that, on average, $O(n^3)$ arithmetic operations are needed to obtain a solution. The problem of missing and spurious points in the images is also briefly discussed. A short summary and examples from simulated data experiments are given.