NAVIGATING IN MOVING INFLUENCE FIELDS

by

Y. Pnueli, N. Kiryati, A.M. Bruckstein

Technical Report #599

December 1989
NAVIGATING IN MOVING INFLUENCE FIELDS

Y. Pnueli †, N. Kiryati ‡ and A. M. Bruckstein †

† - Computer Science Department  ‡ - Electrical Engineering Department
Technion, Israel Institute of Technology
Haifa, Israel 32000

ABSTRACT

This paper applies existing Hough transform methods used for detecting straight lines [2] and general two dimensional shapes [3] in images and for straight line fitting [4] to problems of constant velocity motion design in the presence of moving objects in two and three dimensions. Hough transform methods solve optimization problems in two stages. In the first, accumulation stage, the cost of each possible solution is calculated. In the second, search stage, the parameter space is searched for the optimal solution. We show that in the appropriate parameter space, the locus of velocities incurring the same cost by the encounter with a given moving object is a straight line passing through the origin. We utilize this fact to achieve an efficient accumulation algorithm. Since the computational complexity of the algorithm is dominated by the accumulation stage this results in an overall efficient algorithm.

1. THE CONSTANT VELOCITY FRIEND AND FOE PROBLEM

Suppose a spaceship wants to cross a region of space where various objects or targets, friends and foes, move with constant velocities. Each target is surrounded by an "influence" or "cost" field describing the reward or penalty of approaching it to a minimum distance of \( r \). The spaceship wants to cross the region at a constant velocity and has complete knowledge on the motion and location of the friends and foes moving around. The question is how to choose the velocity so that the total cost of the spaceship trajectory, equal to the sum of the costs incurred by the encounter with each target in the region is be minimized. More formally: Let \( XY \) be a two dimensional space populated by \( n \) targets, \( a_1, a_2 \ldots a_n \), and a spaceship \( b \). Each target \( a_i \) has an initial location vector \( \vec{x}_i \), a velocity vector \( \vec{v}_i \) and a cost field \( c_i \) which is a function of \( r_{\text{min}} \) - the minimum distance between \( a_i \) and the spaceship. The spaceship is located at some point \( \vec{x}_b \) and the problem is to find a velocity vector \( \vec{v}_b \) which optimizes the cost function \( \sum_{i=1}^{n} c_i (r_{\text{min}}) \). In this paper we present an algorithm which solves this, so called, constant velocity friend and foe problem. We also show how variations of the algorithm solve related problems in which the cost functions depend on timing as well as on the distance between the spaceship and the targets.
2. WHY NOT TRY AN ANALYTIC SOLUTION?

For a specific set of targets and cost functions an analytic solution can be found provided the cost functions are differentiable with respect to the velocity vector $\mathbf{v}_b$. When the cost functions are only piecewise differentiable, an analytic solution can still be attempted. In this case, however, we have to find a specific solution for every separate range. The overall solution is chosen to be the specific solution which minimizes the cost function among the solutions in the different ranges.

Let $(v_{x_a}, v_{y_a})$ be the components of $\mathbf{v}_b$. Using an analytic method we have to solve the following set of two equations:

$$\frac{\partial \sum c_i (r_{\text{min}})}{\partial v_{x_a}} = 0 \implies \sum \frac{\partial c_i (r_{\text{min}})}{\partial v_{x_a}} = 0 \implies \sum \frac{dc_i}{dr_{\text{min}}} \frac{\partial r_{\text{min}}}{\partial v_{x_a}} = 0 \quad (1)$$

$$\frac{\partial \sum c_i (r_{\text{min}})}{\partial v_{y_a}} = 0 \implies \sum \frac{\partial c_i (r_{\text{min}})}{\partial v_{y_a}} = 0 \implies \sum \frac{dc_i}{dr_{\text{min}}} \frac{\partial r_{\text{min}}}{\partial v_{y_a}} = 0 \quad (2)$$

The distance between a given target $a_i$ and the spaceship at a given moment $t$ is given by

$$r(t) = \sqrt{(X_b(t) - X_i(t))^2 + (Y_b(t) - Y_i(t))^2}$$

Substituting explicit expressions for $X_b(t), X_i(t), Y_b(t)$ and $Y_i(t)$ we get

$$r(t) = \sqrt{(x_b + v_{x_a} t - (x_i + v_{x_i} t))^2 + (y_b + v_{y_a} t - (y_i + v_{y_i} t))^2} \quad (3)$$

The time $r_{\text{min}}$ is reached is found by

$$\frac{d(r^2)}{dt} = 0 \implies 2r((v_{x_a} - v_{x_i})^2 + (v_{y_a} - v_{y_i})^2) + 2(x_b - x_i)(v_{x_a} - v_{x_i}) + 2(y_b - y_i)(v_{y_a} - v_{y_i}) = 0$$

Solving for $t$

$$t = \frac{(x_i - x_b)(v_{x_a} - v_{x_i}) + (y_i - y_b)(v_{y_a} - v_{y_i})}{(v_{x_a} - v_{x_i})^2 + (v_{y_a} - v_{y_i})^2} \quad (4)$$

The value of $r_{\text{min}}$ is found by substituting $t$ from (4) to equation (3).

$$r_{\text{min}} = \frac{1}{\sqrt{(v_{x_a} - v_{x_i})^2 + (v_{y_a} - v_{y_i})^2}} ((x_b - x_i)(v_{y_a} - v_{y_i}) - (y_b - y_i)(v_{x_a} - v_{x_i}) \quad (5)$$

Note, that since we are only interested in the minimum distance between spaceship and target from time $t = 0$ on, equation (5) gives the correct value of $r_{\text{min}}$ only if $t$ in equation (4) is non-negative. Equation (4) gives a negative value of $t$ when the velocities of the spaceship and a target are such that the spaceship is moving away from the target from time $t = 0$. In that case the minimum distance is found by

$$r_{\text{min}} = r_{t=0} = \sqrt{(x_b - x_i)^2 + (y_b - y_i)^2}$$

Therefore, $r_{\text{min}}$ is only piecewise differentiable with respect to $\mathbf{v}_b$.

Consider any cost function which is not a constant. Even if the cost function is differentiable, over its entire range, with respect to $r_{\text{min}}$, it is only piecewise differentiable with respect to $\mathbf{v}_b$. Therefore, even for a well behaved cost function, we have to consider solutions in two separate ranges. Since the two ranges, in general, do not fully overlap for

September 24, 1989
different targets, when we have \( n \) targets, we can expect \( O(n^2) \) different ranges in which we would have to calculate a solution. In general, equations (1) and (2) do not admit an analytic solution; however a numerical solution for these equations can be attempted. When the number of targets is large or when the cost functions are not differentiable over their entire support, the number of ranges in which we have to numerically calculate a solution increases and this method looses its appeal. We therefore attempt to solve the problem using a different computational approach.

3. THE HOUGH TRANSFORM APPROACH

The Hough transform [1] was originally devised as a technique for detecting complex patterns in bubble chamber images. A standard application is the detection of straight lines in digital images, for example, using normal parameters as suggested by Duda & Hart [2]. Ballard [3] generalized the algorithm to allow the detection of arbitrary shapes. In a recent paper Kiryati & Bruckstein [4] extended the Duda & Hart algorithm to detect the optimal straight path between stationary points that are surrounded by radially symmetrical influence fields. The present research extends the results of [4] to the case in which these points are moving with a constant velocity.

To solve the constant velocity friend and foe problem using a parameter space (Hough) transformation method we do the following: First, we create a discrete parameter space of possible values of \( v_b \). Then, we transform the cost function to this parameter space (i.e. we calculate for each point in the parameter space the value of the cost function if the velocity vector corresponding to this point is chosen). This stage is called the accumulation stage. Last, we perform a search stage in which we search for the point in the parameter space having the lowest cost. This point is our desired solution. Clearly, the difficult part is transforming the cost function to the velocity vector space. We do this by associating with each discrete point in the parameter space, a cost accumulator which is initially set to zero. In the accumulation stage, we calculate, for each such point and for every target, the minimum distance between the spaceship and the target if the spaceship were to travel with the corresponding velocity. From this we calculate the cost of the target for the given point. We then add (accumulate) this value to the cost accumulator associated with the velocity point considered.

The main computational burden of the above strategy is calculating the cost for each point and target in the accumulation stage. Suppose we allow \( v_b = (v_x, v_y) \) to vary in a circular discrete zone such that \( \Delta v_x = \Delta v_y = \Delta \) and such that \( v_x^2 + v_y^2 \leq \frac{v_{\max}^2}{\Delta} \) for some given value \( v_{\max} \). In this case the search stage would take \( O((\frac{v_{\max}}{\Delta})^2) \) comparisons. The accumulation stage would take \( O(n \times (\frac{v_{\max}}{\Delta})^2) \) additions plus the complexity of computing at each point the value of the cost function. Using a naive method of computing \( c_i(r_{\min}) \) this would add \( O(n \times (\frac{v_{\max}}{\Delta})^2) \times Z \) operations, where \( Z \) is the cost of computing the cost function for a single velocity value. (Z includes 1 square root operation, 3 divisions, 6 multiplications, 1 addition and 2 subtractions plus the cost of computing \( c_i \) for a given \( r_{\min} \)). To decrease the computational burden, we utilize the special geometry of the \( V_x, V_y \) parameter space (as described in following sections). This reduces the number of computations, for calculating the costs for all velocities, to \( O(n \times \frac{v_{\max}}{\Delta}) \) computations.

September 24, 1989
Thus the total complexity of the algorithm is $O(n \times (\frac{v_{\text{max}}}{\Delta})^2)$ additions and comparisons plus $O(n \times \frac{v_{\text{max}}}{\Delta})$ additional floating point operations.

4. ACCUMULATING THE COST OF A SINGLE TARGET

Suppose we wish to solve the the constant velocity friend and foe problem in two dimensions for a single target $a$ using the above strategy. To make our computations easier we transform the problem to the coordinate system $XY$ which is located on the moving spaceship $b$. In this coordinate system the target is initially located at $x_a = x_a - x_b$, $y_a = y_a - y_b$, and is moving with a velocity of $\vec{v}_x = v_{a_x} - v_{b_x}$, $\vec{v}_y = v_{a_y} - v_{b_y}$. (Having solved the problem in this coordinate system the transformation to a solution in the $XY$ or "world" coordinate system is simple).

In the $XY$ coordinate system the target moves along a straight line given by the equation

$$y - y_a = (x - x_a) \frac{\vec{v}_y}{\vec{v}_x} \implies y = x \alpha + \beta$$

where

$$\alpha = \frac{\vec{v}_y}{\vec{v}_x}$$

and

$$\beta = y_a - x_a \alpha.$$

The distance between a point $(x, y)$ on this line and the origin is given by

$$r = \sqrt{x^2 + y^2} = \sqrt{x^2 + x^2 \alpha^2 + 2x \beta + \beta^2}$$

The minimum distance between the target and the spaceship, which is the minimum distance between this line and the origin, is found by

$$\frac{d(r^2)}{dx} = 0 \implies x = -\frac{\beta}{1 + \alpha^2}$$

thus

$$r_{\text{min}} = \frac{\beta}{\sqrt{\alpha^2 + 1}} = \frac{y_a - x_a \alpha}{\sqrt{\alpha^2 + 1}}$$

Given a candidate velocity $\vec{v}_x, \vec{v}_y$ we can calculate $r_{\text{min}}$ and hence the cost of the target for this velocity. However, note that the cost of the target is the same for any velocity vector $\vec{w}$ such that $\alpha = \frac{\vec{w}_y}{\vec{w}_x} = \frac{\vec{v}_y}{\vec{v}_x}$. In other words, in the parameter space $\vec{V}_x, \vec{V}_y$, corresponding to the coordinate system $XY$, the locus of all velocities resulting in the same minimum distance between target and spaceship, is a straight line passing through the origin (see Figure 1.). We can, therefore, calculate the cost of the target for the entire $\vec{V}_x, \vec{V}_y$ parameter space by performing only $O(\frac{v_{\text{max}}}{\Delta})$ computations of $r_{\text{min}}$ and $c(r_{\text{min}})$.

September 24, 1989
5. ACCUMULATING THE COST FOR MANY TARGETS

Given the above solution for a single target, we solve the multiple target case in the following way: For each target we compute the cost for every candidate vector velocity in the $V_x, V_y$ parameter space, corresponding to the $XY$ coordinate system of that target. We then transform these costs to the $V_x, V_y$ parameter space, which corresponds to the $XY$ "world" (or common) coordinate system. We add each cost to the corresponding accumulator, which stores the total cost of all the targets considered so far. The transformation from the $V_x, V_y$ parameter space to the $V_x, V_y$ parameter space is simple: All we do is rotate the coordinate system 180 degrees and move the origin to $v_x, v_y$ (see Figure 2).

Of course, in practice we accumulate the cost of each target directly to velocity vectors in "world" coordinates. That is we calculate the cost of a given velocity vector $v_x, v_y$, and then add this value to all the velocity points $(w_x, w_y)$, in the $V_x, V_y$ parameter space, located on the straight line

$$w_y - v_y = (w_x - v_x) \times \frac{v_y}{v_x}$$

6. PAST AND FUTURE CONCERNS

Consider Figure 1. again. All the velocity vectors, corresponding to points on a straight line passing through the origin, cause the target to move on the same straight line with respect to the spaceship. However, some of the values cause the target to move toward the spaceship, while others cause it to move away from the spaceship. Clearly, if some value $v_x, v_y$ causes the target to move toward the ship, the value $-v_x, -v_y$ causes it to move away from the ship.

Consider, in the $V_x, V_y$ parameter space, the line $\bar{w}_y = -\frac{x_a}{y_a} \bar{w}_x$ which passes through the origin. This line is the locus of all the relative velocity values $(\bar{w}_x, \bar{w}_y)$ for which $r_{t=0} = r_{\text{min}} = \sqrt{x_a^2 + y_a^2}$. Since for these velocities the minimum is reached at time $t = 0$.

September 24, 1989
the same $r_{\min}$ is incurred, regardless of the sign of $\vec{w}_x, \vec{w}_y$. It is easy to see that if we divide the parameter space along this straight line, all the velocity values on one side cause the target to move away from the ship and all the values on the other side cause the target to move toward it.

Therefore, when we accumulate the cost for each velocity value, we actually have to accumulate a calculated value only for half the parameter space. In the other half, we add the constant cost $c_i(r_{r=0})$.

7. ACCURACY CONCERNS

Because of the discretization of the parameter space, we necessarily loose accuracy. We assume that all velocity values between some $(v_x-\Delta, v_y-\Delta)$ and $(v_x+\Delta, v_y+\Delta)$ cause the spaceship to approach a target to within the same value of $r_{\min}$. By selecting the fineness of the velocity grid ($\Delta$) we can, in general, get an accuracy to within any desired bounds. The only exception is the region surrounding the origin of the parameter space $\vec{V}_x, \vec{V}_y$, where all the "equal cost" lines converge to a single point.

At first glance, this might seem, a serious limitation on the accuracy. The optimal velocity might be located near a point in the $V_x, V_y$ parameter space which is the origin of the $\vec{V}_x, \vec{V}_y$ parameter space of some target. Then a slight variation of the velocity might modify the cost of that target considerably. However, in practice, this problem is never encountered; as we approach the origin, the relative velocity between target and spaceship decreases until, at the origin, it drops to zero. We can therefore find, for every practical purpose, a value $\Delta_{\min}$, such that if the relative velocity between target and spaceship is less than $\Delta_{\min}$, then we can safely assume that the target is not moving with respect to the spaceship.

The parameter space $\vec{V}_x, \vec{V}_y$ corresponding to some target looks like a "porcupine" as shown in Figure 3.

![Figure 3 - A "porcupine"](image-url)
8. EXTENSION TO 3-D

Extending the above method to a three dimensional arena is straightforward. We first solve the problem for a single target by transforming it to the $XYZ$ coordinate system located on the moving spaceship. In this coordinate system the trajectory of the target is a straight line determined by the relative velocity parameters $\vec{v}_x, \vec{v}_y, \vec{v}_z$:

$$y - \bar{y}_a = (x - \bar{x}_a) \frac{\vec{v}_y}{\vec{v}_x} \Rightarrow y = x \alpha + \beta$$

where $\alpha = \frac{\vec{v}_y}{\vec{v}_x}$ and $\beta = \bar{y}_a - \bar{x}_a \alpha$.

$$z - \bar{z}_a = (x - \bar{x}_a) \frac{\vec{v}_z}{\vec{v}_x} \Rightarrow z = x \gamma + \delta$$

where $\gamma = \frac{\vec{v}_z}{\vec{v}_x}$ and $\delta = \bar{z}_a - \bar{x}_a \gamma$.

The distance between a point $(x, y, z)$ on this line and the origin is given by

$$r = \sqrt{x^2 + y^2 + z^2} = \sqrt{x^2 + x^2 \alpha^2 + 2x \alpha \beta + \beta^2 + x^2 \gamma^2 + 2x \gamma \delta + \delta^2}$$

The minimum distance between the target and the spaceship is found by

$$\frac{d(r^2)}{dx} = 0 \Rightarrow x = -\frac{\alpha \beta + \gamma \delta}{1 + \alpha^2 + \gamma^2}$$

substituting for $x$

$$r_{\text{min}} = \sqrt{\frac{(\beta y - \bar{y}_a) + \beta^2 + \delta^2}{\alpha^2 + \gamma^2 + 1}}$$

Again note, that the locus of all velocities resulting in the same minimum distance between spaceship and target is a straight line. We let $\vec{v}_t = (v_x, v_y, v_z)$ vary in a spherical zone such that $\Delta_x = \Delta_y = \Delta_z$ and such that $v_x^2 + v_y^2 + v_z^2 \leq v_{\text{max}}^2$. Note that although the size of the parameter space is $O\left(\left(\frac{v_{\text{max}}}{\Delta}\right)^3\right)$ we have to calculate $r_{\text{min}}$ and $C(r_{\text{min}})$ only $O\left(\left(\frac{v_{\text{max}}}{\Delta}\right)^2\right)$ times and fill in the rest of the parameter space according to straight lines emanating from the origin.

Putting together similar results to sections 6 and 7 above for the accuracy and past/future concerns, we have for each target a three dimensional "porcupine" of "equal cost" lines.

As before, to solve the problem for many targets we transform each porcupine from an $\vec{V}_x, \vec{V}_y, \vec{V}_z$ parameter space of some target, to the $V_x, V_y, V_z$ world parameter space, in which we accumulate the results of all the targets. (Again in practice we accumulate the results of each additional target directly into the accumulators of the $V_x, V_y, V_z$ velocity vector space).

9. THE EXPOSURE-TIME PROBLEM

A natural variation of the constant velocity friend and foe problem is the, so called, two or three dimensional exposure-time problem. The scenario of this problem is the same as of the original problem, except that we associate with each target $a_t$ a cost function $c_t(r(t))$ which is interpreted as the cost incurred by being exposed to the target, for some infinitesimal time $dt$, when the distance between spaceship and target is $r(t)$.
(another interpretation of \( c_i(r(t)) \) is the probable cost of passing at a distance of \( r(t) \) from a target for a time \( dt \). In this case we want to find a velocity vector \( (v_x, v_y) \) which minimizes the sum of the costs incurred by the total time of being exposed to all the targets:

\[
\sum_{i=1}^{n} \int c_i(r(t)) \, dt
\]

Using the parameter space approach, we accumulate the costs for each target separately. Again we move to the \( XY \) or \( XYZ \) coordinate space and there calculate the cost of exposure for each velocity vector in the \( \vec{V}_x, \vec{V}_y, \vec{V}_z \) parameter space. We then transform the results to the world parameter space and accumulate them in the appropriate accumulators.

Because of the dependence of the exposure cost function on the time of exposure, we cannot use the same exposure cost, calculated for some given velocity \( \vec{V}_x, \vec{V}_y \), for all the velocity vectors \( \vec{w}=(w_x, w_y) \) such that \( \frac{w_y}{w_x} = \frac{\vec{V}_y}{\vec{V}_x} \), as we did in the original friend and foe problem. However, we can still save most of the calculations necessary for finding the value of the integral at every point.

Consider two velocity vectors \( \vec{v} \) and \( \vec{w} \) located on the same line passing through the origin of the parameter space \( \vec{V} \) such that \( \vec{p}\vec{w} = \vec{v} \), for some number \( p \). Let

\[
I = \int_{t=0}^{\infty} c_i(r(\vec{v}, t)) \, dt
\]

where \( r(\vec{v}, t) \) is the distance between spaceship and target at time \( t \), when the spaceship travels at a constant velocity \( \vec{v} \). Note that for every velocity \( \vec{v} : \vec{r}(t) = \vec{r}(0) + t\vec{v} \). Therefore

\[
\int_{t=0}^{\infty} c_i(r(\vec{w}, t)) \, dt = \int_{t=0}^{\infty} c_i(r(0) + t\vec{w}) \, dt = \int_{t=0}^{\infty} c_i(r(0) + \frac{t}{p}\vec{v}) \, dt
\]

substituting \( t = pT \) we get

\[
\int_{T=0}^{\infty} c_i(r(0) + T\vec{v}) \, pT \, dT = p \int_{T=0}^{\infty} c_i(r(0) + T\vec{v}) \, dT = pI.
\]

From the above arguments, we see that if we compute the value of the integral for one given velocity, on a ray passing through the origin, then we can find the value of every other velocity on the same ray, by a single multiplication by \( p \) (and a division if we first have to determine \( p \)). If the value of the integral is determined analytically or to a very fine accuracy, then this argument is true for any two such velocity vectors. When the accuracy is limited, it is preferable to always use \( p < 1 \). Therefore, in the accumulation stage of the exposure problem, we fully calculate the value of \( \int_{t=0}^{\infty} c_i(r(t)) \, dt \) for the velocity \( \vec{v} \) with the smallest absolute value on a given ray. We multiply by decreasing values of \( p \) to calculate the expected cost for every other \( \vec{w} \) on the ray.

Note that in the exposure-time problem, we have to calculate the exposure-time cost of every point in the parameter space regardless of whether the spaceship is moving away
from the target or toward it. Therefore for every "line" passing through the origin, we have to fully calculate the exposure-time cost for two velocity values (one moving toward and one moving away from the target) and fill in the results for the remaining velocities on the corresponding rays.

However, note that close to the origin we can still make the approximation that the distance between target and spaceship remains constant. This actually means that we integrate the expected cost not up to $t = 0$ but only up to some fixed time $t = \tau$, however this is not a serious limitation, since when we compute the integral numerically we always bound the integration at some such $\tau$ anyway (assuming the integral converges).

10. THE "BLOW THE MISSILES" PROBLEM

The second variation of the friend and foe problem, which we consider, is the, so called, "blow the missile problem". In this problem, a two or three dimensional region of space is populated by $n$ hostile missiles marked $a_1, a_2, \ldots, a_n$. Each missile $a_i$ is initially located at some point $\vec{x}_i$ and is moving at a constant velocity $\vec{v}_i$. An anti-missile defence base wants to launch a single anti-missile missile, $b$, from some given point $\vec{x}_b$. The missile $b$ travels at a user specified constant velocity $\vec{v}_b$ and can be set to explode at a user specified time. When it explodes, it is guaranteed to destroy everything within a given radius $r$ of itself. The problem is to find a constant velocity vector $\vec{v}_b$ and a time $t$, such that if $b$ is launched at this constant velocity $\vec{v}_b$ and is set to explode at $t$, it will destroy all the hostile missiles in the region (all the hostile missiles in the region will be within a range of $r$ from $b$). When it is impossible to find such a $\vec{v}_b$ and $t$, we say that the problem has no solution (some different variations of this problem are considered at the end of this section).

We solve the problem using a parameter space approach similar to the one used in the previous problems. As before, we create a discrete parameter space of possible values of $\vec{v}$. However, with each point in this parameter space we associate two time accumulators $t_{\text{low}}$ and $t_{\text{high}}$, which are initially set to zero and infinity respectively. These accumulators are used to determine the time range at which $b$ can be set to explode, if the missile travels with the velocity $\vec{v}$.

For every target $a_i$ and every point $\vec{v}$ in the parameter space, we compute $\bar{t}_{\text{low}}$ and $\bar{t}_{\text{high}}$ which are the earliest and latest times at which $a_i$ is within range of $b$, if $b$ is launched with the constant velocity $\vec{v}$. We update $t_{\text{low}}$ to be the maximum of the previous $t_{\text{low}}$ and $\bar{t}_{\text{low}}$. Correspondingly, we update $t_{\text{high}}$ to be the minimum of the previous $t_{\text{high}}$ and $\bar{t}_{\text{high}}$. After all the targets are considered, every point $\vec{v}$ in the parameter space for which $t_{\text{low}} \leq t_{\text{high}}$ is a possible velocity for $b$. If such a velocity $\vec{v}$ is chosen then the explosion time of $b$ should be set to any value between the corresponding $t_{\text{low}}$ and $t_{\text{high}}$.

Clearly, as in the previous problems, the main computational burden of this approach is incurred in the accumulation stage, if we were to compute from scratch $\bar{t}_{\text{low}}$ and $\bar{t}_{\text{high}}$ for every target and every point in the parameter space. We therefore, again, use the geometrical properties of the parameter space to simplify the computations. In the following paragraph, we describe the algorithm for the two dimensional problem. The extension to three dimensions is straightforward.

September 24. 1989
First, for each target, we transform the problem to the $XY$ coordinate space. Let $\mathbf{v} = (v_x, v_y)$ be some candidate velocity for $b$. The path which the target describes in the $XY$ plane is the straight line as in (6)

$$y - y_a = (x - x_a) \frac{v_y}{v_x} \implies y = x \alpha + \beta$$

This straight line intersects a circle of radius $r$ surrounding the origin in the following points:

$$x_1 = \frac{-\alpha \beta + \sqrt{\alpha^2 r^2 + r^2 - \beta^2}}{\alpha^2 + 1}, \quad y_1 = \frac{\beta + \sqrt{\alpha^2 r^2 + r^2 - \beta^2}}{\alpha^2 + 1}$$

and

$$x_2 = \frac{-\alpha \beta - \sqrt{\alpha^2 r^2 + r^2 - \beta^2}}{\alpha^2 + 1}, \quad y_2 = \frac{\beta - \sqrt{\alpha^2 r^2 + r^2 - \beta^2}}{\alpha^2 + 1}$$

From these points, we calculate the times at which the target enters and leaves the range of $b$.

$$t_1 = \frac{x_1 - x_a}{v_x} \quad \text{and} \quad t_2 = \frac{x_2 - x_a}{v_x}$$

The earlier of the two $t$'s is the desired $t_{\text{low}}$ and the later is $t_{\text{high}}$.

Note that although two velocity vectors $\mathbf{v}$ and $\mathbf{w}$ such that $p \mathbf{v} = \mathbf{w}$ do not have the same $t$'s, they do have the same intersection points $(x_1, y_1)$ and $(x_2, y_2)$. We, therefore, have to calculate the intersection points only once for every set of velocities located on the same line in the $V_x, V_y$ parameter space, while we have to calculate the $t$'s separately for each point on the line. Since the computation of the $t$'s is simpler we still have a computational advantage.

A few variations of the blow the missile problem can be solved using slight variations of the above technique:

1. The case in which some of the targets are friends and not foes.
2. The case in which, if we cannot destroy all the missiles, we want to destroy the maximum number of missiles.

In both cases, we consider a discrete parameter space which is composed of the velocity vector $\mathbf{v}$ and the time $t$. When we calculate the $t_{\text{low}}$ and $t_{\text{high}}$ of every target, we accumulate the information of whether the target is within range or not, for every discrete value of $t$. In the first case, where some of the targets may be friends, we just store a yes or no information indicating whether a given parameter set $(\mathbf{v}, t)$ can be used (yes if all the foes and none of the friends, considered so far, are within range, and no otherwise). In the second case, we accumulate the number of foes which we "hit" for every candidate parameter set. When all the targets have been considered, we search for the parameter set with the highest number of "hits".

Note that an extension to the case when we have both friends and foes, each with a cost function indicating the cost of hitting it, is straightforward.

September 24, 1989
11. References


