Computing Robust Viewpoints with Multi-constraints Using Tree Annealing

Yulin Yao*
Peter Allen†

Abstract

In order to compute camera viewpoints during sensor planning, Tarabanis et al present a group of feature detectability constraints which include six nonlinear inequalities in an eight-dimensional real space. It is difficult to compute robust viewpoints which satisfy all feature detectability constraints. In this paper, the viewpoint setting is formulated as an unconstrainted optimization problem. Then a tree annealing algorithm, which is a general-purpose techniques for finding minima of functions of continuously-valued variables, is applied to solve this nonlinear multiconstraint optimization problem. Our results show that the technique is quite effective to get robust viewpoints even in the presence of considerable amounts of noise.

I. INTRODUCTION

Sensor planning involves determining strategies with which sensor parameter values can be found that will achieve a sensing task with a certain degree of satisfaction. It is a fairly new area of computer vision but has received considerable interest recently [2] [6] [7] [8]. Tarabanis, Tsai and Allen [7] have been developing a vision planning system, MVP (Machine Vision Planner), that automatically determines vision sensor parameter values so that the task requirements are satisfied. Compared to the iterative techniques employed in the SRI system [2] and other sensor planning systems, the main contribution of the MVP system is that it provides closed-form solutions to the individual task constraints and determines a set of sensor parameters which characterize the general viewing configurations. However, in the MVP system, it is difficult to compute robust viewpoints which satisfy all feature detectability constraints simultaneously. As Tarabanis pointed out in [6], techniques that combine the admissible domain of individual constraints in order to determine optimal solutions still need to be investigated.

In this paper, the viewpoint setting is formulated as an unconstrainted optimization problem, then a tree annealing (TA) technique [1] which is one of simulated annealing algorithms [4] that can handle continuously-valued variables, is applied to solve the multiple nonlinear constraints problem. Our results show that the technique is quite effective to get robust viewpoints even in the presence of considerable amounts of noise.

II. CONSTRAINTS FOR FEATURE DETECTABILITY

In the MVP system, the configurations of viewing parameters that are planned include the three positional degrees of freedom of the sensor $\vec{r_o}$(x,y,z), the two orientational degree of freedom (pan and tilt

*The author was with the Columbia University. He is now with the Dow Jones & Company, Inc.
†The author is with the Columbia University

angles) described by a unit vector $\vec{v}$ along the viewing direction and the three optical parameters (the back nodal point to image plane distance d, and focal length f and the aperture of the lens a). Thus, planning is done in eight-dimensional space and a point in this space is defined as a *generalized viewpoint* $\vec{V}(\vec{r_o}, \vec{v}, d, f, a)$. Using knowledge from geometry and optics, each task constraint in the MVP is characterized by an analytical relationship [6]. As a result, the locus of generalized viewpoints that satisfies the resolution, depth-of-field and field-of-view constraints separately is expressed by a relationship of the form: $g_i(\vec{r_o}, \vec{v}, d, f, a) \geq 0$, specifically:

•Depth of field:

for farthest point: $g_1 = D_1 - ||(\vec{r_O} - \vec{r_f}) \cdot \vec{v}|| \geq 0$, for closest point: $g_2 = ||(\vec{r_O} - \vec{r_c}) \cdot \vec{v}|| - D_2 \geq 0$

where $\vec{r_f}$ is the position vector of the farthest feature vertex from the front nodal point of the lens along the viewing direction; $\vec{r_c}$ is the position vector of the closest feature vertex from the front nodal point of the lens along the viewing direction; $\vec{r_O}$ is the position vector of the front nodal point of the lens and

$$D_1 = \frac{Daf}{af - c(D-f)},\quad D_2 = \frac{Daf}{af + c(D-f)},\quad D = \frac{1}{(1/f - 1/d)}.$$

where c is the minimum of the horizontal and vertical sensor element spacings, d is the back nodal point to image plane distance, f is the focal length and a is the aperture of the lens.

•Field of view:

$$g_3 = (\vec{r_K} - \vec{r_O}) \cdot \vec{v} - cos(\tfrac{\alpha}{2})||\vec{r_K} - \vec{r_O}|| \geq 0$$

where $\vec{r_K} = \vec{r_C} - R_o\vec{v}$, $\vec{r_C}$ is the position vector of the center of the sphere of radius R_f circumscribing the object features, $R_o = R_f/(sin(\alpha/2))$, R_f is the radius of the sphere circumscribing all the object features, α is the field of view angle and is given by $\alpha = 2 \cdot tan^{-1}(I_{min}/2d)$, I_{min} is the minimum dimension of the sensor plane, and all other variables are as defined above.

•Resolution for edge feature $\overline{AB}_i$:

$$g_4 = \frac{||\vec{v} \times [\vec{e_i} \times (\vec{r_O} - \vec{r_{A_i}})]||}{[(\vec{r_{A_i}} - \vec{r_O}) \cdot \vec{v}][(\vec{r_{B_i}} - \vec{r_O}) \cdot \vec{v}]} - \frac{w}{dl} \geq 0$$

where $\vec{r_O}, \vec{r_{A_i}}, \vec{r_{B_i}}$ are the position vectors of the front nodal point of the lens and vertices of the feature edge i to be resolved; $\vec{e_i}$ is the unit vector along to feature edge $\overline{AB}_i$ to be resolved; l, w are the lengths of the feature to be resolved in object and image space, respectively. All other variables are as defined above.

Unit vector:

$$g_5 = ||\vec{v}||^2 - 1 = 0$$

It should be noted that there is a resolution constraint for each edge feature that is to be resolved, while for other constraints, there is a single relationship for all features.

III. TREE ANNEALING

Tree annealing [1] [3] is an extension of the familiar Metropolis algorithm [5] of simulated annealing, but handles continuously valued variables in a natural way. In this section, we briefly introduce the tree annealing (TA) method based on [1] [3].

Following the definitions and notations of [1] [3], let us assume we are searching for the minimum of some function f($\mathbf{x}$) where the d-dimensional vector $\mathbf{x}$ has continuously valued elements. Furthermore, we assume a finite search space $S \subset R^d$. A k-d tree in which each level of the tree represents a binary partition of one particular degree of freedom (DOF) is used. Each node may thus be interpreted as representing a hyperrectangle, and its children therefore represent the smaller hyperrectangles resulting from dividing the parent along one particular DOF.

Let a vector $\mathbf{x}$ be the current sample. At each node, two numbers are stored, n_L and n_R, representing how many times in the past that an acceptable point has been found in the left and right subtrees, respectively. The TA algorithm works as follows for a finite set S:

1. Growing and searching the tree:

   (a) The tree is initiated by simply creating the root node, and choosing a point at random with uniform probability from the entire search space. That point becomes the first accepted point. Two daughter nodes are created, corresponding to a division of the search space in half along the first DOF. The n_L and n_R are both initialized to 1 for the root node.

   (b) Begin at the root and, at each node, choose either the left or right child randomly with probability $\frac{n_L}{n_L+n_R}$ or $\frac{n_R}{n_L+n_R}$ respectively. Descend the tree to its leaves making left-right decisions in this way.

   (c) Upon reaching a leaf, generate the point $\mathbf{y}$ at random (uniformly) from the subspace defined by the leaf. Compare $\mathbf{x}$ and $\mathbf{y}$ and make an accept/reject decision on $\mathbf{x}$ (see step 2). If $\mathbf{y}$ is accepted, replace $\mathbf{x}$ by $\mathbf{y}$ as the current sample; if $\mathbf{y}$ is rejected, $\mathbf{x}$ remains the current sample.

   (d) If $\mathbf{y}$ was accepted, split the current leaf (containing $\mathbf{y}$), and create two new daughter nodes, thus making more resolution available at this node if it is ever explored again.

   (e) Ascend the tree from the current sample to the root, updating n_L and n_R at each node.

2. Accept/reject decision. Accept the point $\mathbf{y}$ as the new estimate with probability

$$min(1, \frac{g(\mathbf{x})}{g(\mathbf{y})}\frac{p(\mathbf{y})}{p(\mathbf{x})})$$

   where the probabilities p are Gibbs, (i.e., with form $p(\lambda) \propto exp(-\frac{f(\mathbf{x})}{T})$) and g($\mathbf{y}$) is computed from the path of the descent down the tree by

$$g(\mathbf{y}) = \frac{1}{V_y}\prod_l p_l'$$

   where $p_l' = \frac{a_l}{a_l+b_l}$, l represents the node visited at level l, a_l represents n_L or n_R, according to which direction was chosen ar each l. Similarly, b_l represents the n of the direction not chosen.

3. The annealing schedule is very similar to the suggestion in [4]:

$$T \leftarrow rT$$

   where r = $1 - \frac{dS}{C_v}$ and dS is a small positive constant and C_v is a term easily related to the variance of the energy.

IV. PROBLEM FORMULATION

The decisive criteria of a computed viewpoint are its robustness and stability. The measure is used to assess the goodness of a solution with respect to the value of each constraint relationship g_i. This is appropriate since a large positive value of g_i indicates that a constraint is satisfied comfortably, a small positive value indicates marginal satisfaction, while inadmissible solutions give rise to a negative value. We want to search a globally admissible eight-dimensional viewpoint which is near the center of the admissible domain and far from the bounded hypersurfaces described by the constraints. Such a generalized viewpoint is desirable, since it is robust in the event of inaccuracy. Similarly, the measure for the visibility constraint is also formulated. For this purpose, the minimum distance, d_v, from the viewpoint to the polyhedron describing the visibility region is chosen: $g_6 = \pm d_v$, where $+d_v$ or $-d_v$ depending on whether the point is inside or outside the visibility volume respectively. The optimization function is taken to be a weighted sum of the above component criteria, each of which characterize the quality of the solution with respect to each associated requirement separately. If we take two edge features then we will have two resolution constraints g_{4a} and g_{4b}, each of them with respect to an edge feature. Thus, the optimization function is written as:

$$\max_S obj = \sum_i \alpha_i \cdot g_i,\ (i = 1,2,3,4a,4b,6), or \min_S obj = -\sum_i \alpha_i \cdot g_i,\ (i = 1,2,3,4a,4b,6)$$

Subject to:

$g_i \geq 0$; i = 1, 2, 3, 4a, 4b, 6; and $g_5 = 0$

where α_i are weights and s is a point of the finite eight-dimensional space S.

We convert the above set of constraints into a penalty function. For each $g_i, (i = 1,2,3,4a,4b,6)$, the penalty term $exp(-\beta_i g_i)$ is assigned, where β_i is a positive real number which represents the degree of penalty (penalty factor). It is appropriate since, for $g_i < 0$, the value of the $exp(-\beta_i g_i)$ will be (exponentially) very large; for $g_i \geq 0$, the value of the $exp(-\beta_i g_i)$ will be small. For g_5, the penalty term $exp(\beta_5|g_5|)$ is assigned, where β_5 is a positive penalty factor. In our experiments, we choose same penalty factor (=1) for each constraint g_i, (i =1, 2, 3, 4a,4b). It is also appropriate since, for $g_5 = 0$, $exp(\beta_5|g_5|) = 1$; for $g_5 \neq 0$, the value of $exp(\beta_5|g_5|)$ will be (exponentially) very large. In our experiments, we choose $\beta_5 = 1000$, which is larger than any other penalty factor, in order to get more accurate unit vector. So we know that the penalty function will appropriately penalize any infeasible/inadmissible constraint. Thus the constrained problem is reformulated as an unconstrainted optimization:

$$\min_{s \in S} obj = -\sum_i \alpha_i \cdot g_i + \sum_i exp(-\beta_i \cdot g_i) + exp(\beta_5 \cdot |g_5|)$$

where i = 1, 2, 3, 4a, 4b, 6.

We use the TA algorithms described in the previous section to solve this unconstrainted optimization.

V. EXPERIMENTAL RESULTS

As part of the MVP system, we have implemented the vision planning algorithms that are given in Section II and III using the TA algorithm. In the experiments, we will demonstrate the effectiveness of applying the technique to compute the robust general viewpoints with multiple feature detectability constraints. The features to be observed are the two edges (a and b) of an enclosed cube.

In our experiments, we choose the parameters as in [6]: $\vec{r_O}$= (0,0,0), which coincides with the origin of object coordinates system; c = 13.5 microns, l = 2.54 mm, w = 0.02112 mm, I_{min} = 6.5 mm, where

c is the minimum of the horizontal and vertical sensor element spacings, l, w and I_{min} are defined in Section II. The values of the lens aperture a and the intrinsic focal length f are chosen a priori ($f = 12.5$ mm and $a = f/16 = 0.78125$ mm) and thus, values for the remaining imaging space parameters $\vec{v}$ and d are computed. All measured units are expressed in millimeters in the experiments. The values of the weights α_i in the objective function are taken to be: $\alpha_1 = 0.1$, $\alpha_2 = \alpha_3 = 0.01$, $\alpha_{4a} = \alpha_{4b} = 1000$. The values of the weights β_i in the penalty function are taken to be: $\beta_1 = \beta_2 = \beta_3 = \beta_{4a} = \beta_{4b} = \beta_6 = 1$, $\beta_5 = 1000$ (explained in previous section).

An initial viewpoint V_i that is chosen to start the optimization, and the corresponding camera viewpoint V_f that is computed by the TA algorithm, is listed in Tables 1. For V_i, the g_1 and g_3 constraints are violated (refer to the first column in the Tables 2). All feature detectability of constraints of the computed viewpoint V_f that is determined by the TA algorithm are satisfied.

Table 1: The initial and final generalized viewpoints V_i and V_f (unit: mm)

	x	y	z	v(1)	v(2)	v(3)	f	d	a
V_i	80.0	-5.0	160.0	-0.58	0.2	-0.8	12.5	14.0	f/16
V_f	124.34	-4.44	207.60	-0.61	-0.03	-0.79	12.5	13.13	f/16

In order to check the robustness and stability of the computed viewpoints, the camera is approached to the object along the view direction to see whether the constraints are still satisfied. Let P_1 be the computed viewpoint, P_2 be a point at which the camera approaches the object and C be the center of the sphere of circumscribing the object features. The approaching scale factor is defined as follows:

$$\text{scale factor} = \frac{|P_1P_2|}{\text{the projection of } \vec{P_1C} \text{ on the view direction} \vec{P_1P_2}}$$

The value of constraints with different scale factors are given in Tables 3 and 4. We find all constraints but g_2 and g_3 are satisfied. The constraint g_2 (depth of field for closest point) is isolated when the distance between the viewpoint and the center of the object is less than the certain value (D_2); and the constrain g_3 (focus of view) is isolated when the angle between the view direction $\vec{P_1P_2}$ and $\vec{P_2C}$ is greater than certain value. These values are determined by the intrinsic parameters of camera (see the definition of g_2 and g_3 in section II). The interesting result — the maximum reachable viewpoint V_{max}, which still satisfy simultaneously all constraints, from the current computed viewpoint V_f respectively along a reverse view direction — is given in the third column in Table 2.

Another factor that will affect the stability and robustness of the computed viewpoint is the presence of noise, for example, the slight perturbation of manipulator on which the camera is mounted (we can imagine that the manipulator is teleoperated and many conditions around it are unpredictable). In order to check the stability and robustness of viewpoint planning in the presence of noise, independent random noise with 10 %, 20 % and 30 % are added to each component of the position vector $\vec{r_o}$ and the orientation vector $\vec{v}$. The values of constraints under the different noise levels is listed in Tables 2. We can find from the table that all constraints are still satisfied in these cases, that is, the computed viewpoint V_f are stable and robust even in the presence of noise. Thus we can conclude that the viewpoint V_f which is computed by the TA algorithm is robust and stable.

ACKNOWLEDGMENTS The authors would like to thank Steven Abrams and Paul Michelman for their valuable help and comments.

Table 2: The values of constraints with the different scale factors and noises for V_i, V_f and V_{max}(unit: mm)

				scale factor			noise		
	V_i	V_f	V_{max}	0.10	0.25	0.50	0.10	0.20	0.30
x	80.00	124.34	215.23	109.63	87.57	50.80	135.57	140.48	134.73
y	-5.0	-4.44	-9.62	-3.60	-2.35	-0.25	-4.45	-5.17	-4.91
z	160.00	207.60	324.96	188.61	160.13	112.66	219.39	209.41	225.02
v(1)	-0.58	-0.61	-0.61	-0.61	-0.61	-0.61	-0.65	-0.69	-0.67
v(2)	0.20	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04
v(3)	-0.80	-0.79	-0.79	-0.79	-0.79	-0.79	-0.82	-0.87	-0.93
f	12.50	12.50	12.5	12.50	12.50	12.50	12.50	12.50	12.50
d	14.00	13.13	13.13	13.13	13.13	13.13	13.13	13.13	13.13
a	f/16	f/16	f/16	f/16	f/16	f/16	f/16	f/16	f/16
g_1	-42.98	150.36	1.83	174.39	210.44	270.52	134.73	139.54	128.62
g_2	52.09	28.75	177.28	4.71	-31.33	-91.42	44.58	39.74	49.99
g_3	-8.53	1.55	7.22	0.34	-1.98	-11.17	2.02	3.26	3.52
g_{4a}	0.0047	0.0031	0.0016	0.0035	0.0045	0.0076	0.0028	0.0028	0.0027
g_{4b}	0.0021	0.0016	0.0008	0.0018	0.0022	0.0031	0.0015	0.0017	0.0014

References

[1] Griff Bilbro and Wesley Snyder. Optimization of functions with many minima. *IEEE Transactions on Systems, Man and Cybernetic*, 21(4):840–849, 1991.

[2] C.K. Cowan and A. Bergman. Determining the camera and light source location for a visual task. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 509–514, 1989.

[3] Younsik Han, Griff Bilbro, and Wesley Snyder. Pose determination using tree annealing. In *Proceedings of IEEE International Conference on Robotics and Automation*, Cincinnati, OH, 1990.

[4] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[5] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Jouranl of Chemical Physics*, 21(6):1087–1091, 1953.

[6] K. Tarabanis. *Automatic synthesis of camera views that satisfy feature detectability constraints*. PhD thesis, Computer Science Department, Columbia University, New York, NY, 1991.

[7] K. Tarabanis, R. Y. Tsai, and P. Allen. Automated sensor planning for robotic vision tasks. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1991.

[8] S. Yi, R. M. Haralick, and L. G. Shapiro. Automatic sensor and light source positioning for machine vision. In *Proceedings of IEEE International Conference on Pattern Recognition*, pages 55–59, 1990.