Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

A Method of Automatic Sensor Placement for Robot Vision in Inspection Tasks

S. Y. Chen and Y. F. Li
Department of Manufacturing Engineering and Engineering Management,
City University of Hong Kong, Kowloon, Hong Kong
Email: sy@ieee.org, meyfli@cityu.edu.hk

Abstract

This paper presents an automatic sensor placement technique for robot vision in inspection tasks. In such vision systems, a sensor often needs to be moved from one pose to another around the object to sample all features of interest. Multiple 3D images are taken from different vantage points. The technique involves deciding the optimal sensor placements and a shortest path through these viewpoints for automatic generation of an inspection plan. A viewpoint is expressed by N parameters and a topology of viewpoints is achieved by genetic algorithm. The inspection plan is evaluated using a min-max criterion and the shortest path is determined by Christofides algorithm. In addition, a computation example is presented to illustrate the techniques and algorithms.

1. Introduction

With the rapid growth of automation in the manufacturing industry, computer vision now plays an important role for inspection, assembly, recognition and reverse engineering, etc. Since a vision sensor can only sample a portion of an object from a single viewpoint, multiple 3D images need to be taken and integrated from different vantage points to enable all features of interest to be measured. Sensor placement thus plays a significant role in achieving an economic planning strategy, which determines the subsequent viewpoints and offers the obvious benefit of reducing and eliminating the labor required.

Sensor placement has been studied for more than ten years. It can be classified into two application categories, model based and non-model based. Typical non-model based application is 3D object reconstruction ([1, 2, 3, 4]) and model based applications are widely used in inspection, recognition, and assembly etc. ([5, 6, 7, 8, 9]). Previous approaches to sensor placement are mainly focused on modeling of sensor constraints and calculating a "good" viewpoint to observe one or several features on the object. It is usually not considered to achieve the overall efficiency of a generated task with a sequence of viewpoints. This paper is dedicated to developing a robust method for planning inspection tasks with both viewpoint topology and sensing sequence. In such tasks, the procedure of plan generation is described as:

(a) Input the object's geometrical information from CAD models;
(b) Given the specifications of the inspection tasks;
(c) Generate a sensor placement graph with least viewpoints;
(d) Search a shortest path for robot operation; and
(e) Output the inspection plan.

In a word, the problem of sensor placement for inspection is to search an optimal topology of placement graph and a shortest path for performing the sensing operations. In this paper, the geometrical information of the object is loaded from a 3-D CAD data file. A strategy is developed to automatically determine a group of viewpoints for a specified vision-sensor with several placement parameters such as position, orientation, and optical settings. Each viewpoint should satisfy many constraints due to some physical and optical properties of the sensor, scene occlusion, and robot reachability in the environment. The sensing plan is evaluated by a min-max criterion, which is achieved by a hierarchical genetic algorithm (HGA), and the shortest path for robot moving through the viewpoints is determined by Christofides algorithm. Combining the two algorithms will give a complete solution of the model-based sensor placement problem.

2. Cost Evaluation of Sensor Placement Plan

2.1 Previous Approaches

In most related work, limitations of sensor placement are expressed as a cost function with the aim to reach the goal with minimal cost. This cost function should have a value tending to infinity associated to the direction of a pose that the sensor cannot assume and a unitary value associated to the direction of a pose that is possible for the sensor to assume [5].

Banta etc. [3] defined the term "best-next-view" (BNV) as the next sensor pose which will acquire the greatest amount of previously unseen three-dimensional information.

[6] [9] chose to formulate the probing strategy as a function minimization problem. The optimization function is taken to be a weighted sum of several component

criteria, each of which characterizes the quality of the solution with respect to each associated requirement separately. Thus the optimization function is written as:

$$h = \max(a_1 g_1 + a_2 g_2 + a_3 g_3 + a_4 g_4) \qquad (1)$$

subject to $g_i \geq 0$, standing for satisfying four constraints, i.e. resolution, focus, field-of-view, and visibility. (In this section the equations are not be explained in detail because of the limitation of space, interested readers please refer to original source).

In [1], the strategy of viewpoint selection takes into account three problems: quality of a new position, displacement cost, and additional constraints. More precisely, it includes: 1) the new observed area volume $G(f_{t+1})$, 2) the cost function F in order to reduce the total camera displacement $C(f_t, f_{t+1})$, and 3) constraints to avoid unreachable viewpoints and to avoid positions near the robot joint limits $B(f)$. The cost function F_{next} to be minimized is thus defined as a weighted sum of the different measures:

$$F(f_{t+1}) = A(f) + a_1 g(f_{t+1}) + a_2 C(f_t, f_{t+1}) + a_3 B(f)\ . \qquad (2)$$

[2] evaluate the suitability of all potential viewpoints of the NBV by using a rating function as

$$f(q, f) = w_e f_e(q, f) + w_o f_o(q, f) + w_s f_s(q, f) \qquad (3)$$

where q and f are two parameters on the viewpoint sphere; fe, fo, fs are factor functions rating on some physical or heuristic constraints, and w_e, w_o, w_s are weighting coefficients. The viewpoint of the largest value of $f(q,f)$ will be chosen as the NBV.

Ye et al. [8] addressed that the total cost for applying the searching effort allocation is:

$$T[F] = \sum_{i=1}^{k} t_o(f_i) \qquad (4)$$

where the cost $t_o(f)$ gives the total time needed to manipulate the hardware to the status specified by f , to take a picture, to update the environment and register the space, and to run the recognition algorithm. The effort allocation $F=\{f_1, ..., f_k\}$ gives the ordered set of operations applied in the search. And the probability of detecting the target by the allocation is:

$$P[F] = P(f_1) + ... + \{\prod_{i=1}^{k-1}[1 - P(f_i)]\}P(f_k) \qquad (5)$$

where $P(f)$ is the probability of detecting the target.

Then the next action is selected that maximizes the term

$$E(f) = \frac{P(f)}{\Delta_T(f)}, \qquad \Delta_T(f) = t_o(f)\ . \qquad (6)$$

Triggs et al [4] gave a method of the function optimization technique to minimize their viewpoint evaluation function. They divide the search space into a set of local regions and build a probabilistic function interpolation or subjective probability distribution for the function value. These distributions can be used to choose which region to refine and where to subdivide it. The goal is to optimize the function, so a sample only "succeeds" if it improves on the best currently known function value f_{best}. If the probability density for the function value at some point is $p(f)df$, the expected gain or improvement to f_{best} from a sample placed at that point is

$$< gain >= \int_{-\infty}^{f_{best}} (f_{best} - f)\, p(f) df \qquad (7)$$

We may find that all these previous methods for solving sensor placement problem are with straight-forward representations. They determine that if a viewpoint is admissible in the space by direct computation. However, the large computation complexity results in a heavy burden to the vision system because there are so many constraints should be satisfied. Therefore these methods usually can not give an efficient solution for a general task. In this paper, we also minimize the cost, but by evolutionary computing, so that it is robust for treating with many different vision tasks and the objects and sensor parameters can be given by the user just at the beginning of computation. Furthermore, a shortest path for robot execution is also suggested by graph theory.

2.2 Lowest Travelling Cost

In this paper, considering there is a priori model of the object, the procedure for generating a sensor placement plane is described as:

(a) Generate a number of viewpoints.
(b) Construct a graph corresponding to the topology of viewpoints. If it satisfies all placement constraints, go to step (d); else increase the number of viewpoints.
(c) Reduce redundant viewpoints.
(d) Compute the lowest cost to optimize robot operations.

Generating a large number of viewpoints will most likely satisfy all constraints and finish the vision task, but it will also increase the cost. To achieve an optimal solution, we need eliminate all possible redundant viewpoints. Fig. 1 illustrates that the 2nd viewpoint is redundant because it does not increase any information of the object model.

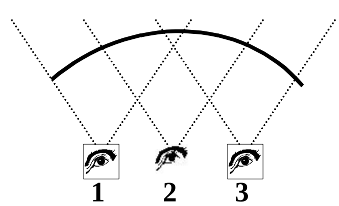

Fig. 1 A redundant viewpoint

A plan of viewpoints is mapped into a graph $G=(V(G),E(G),y_G,w_E)$ with weight $\boldsymbol{w}$ on every edge $\boldsymbol{E}$, where vertices $\boldsymbol{V}_i$ represent viewpoints. Edge $\boldsymbol{E}_{ij}$ represent a shortest collision-free path between viewpoint $\boldsymbol{V}_i$ and $\boldsymbol{V}_j$, and weight $\boldsymbol{w}_{ij}$ represent its corresponding distance. Such a graph is termed as ***sensor placement graph G*** in this paper.

Fig. 2 illustrates an example topology of viewpoint plan. A practical solution of sensor placement problem must provide the exact number of viewpoints which are reachable to the robot and there must exist a collision free path between every two acceptable viewpoints.

A sensor placement graph $\boldsymbol{G}$ has characteristics:

(a) $\boldsymbol{G}$ is a simple undirected graph, i.e. there are no loops and no paralleled edges;

(b) $\boldsymbol{G}$ is a connected graph, i.e. there exists at least one path from vertex $\boldsymbol{V}_i$ to $\boldsymbol{V}_j$;

(c) $\boldsymbol{G}$ is a complete weighted graph, i.e. every pair of vertices $\boldsymbol{V}_i$ and $\boldsymbol{V}_j$ is directly connected with a weight;

(d) $\boldsymbol{G}$ is a finite nontrivial graph, i.e. $1<o(G),\partial(G)<\infty$;

(e) The order and the size of $\boldsymbol{G}$ are $o(G)=n$, $\partial(G)=\frac{1}{2}n(n-1)$, respectively;

The shortest path for taking all views is a Hamilton cycle which is a sequence of vertices: $\boldsymbol{C}=(x_1, x_2, \ldots, x_n, x_1)$ where $xi \neq xj, xi \in V(G), i \in [1,n]$. The path length is

$$l_c = w(x_n,x_1)+\sum_{i=1}^{n-1} w(x_i,x_{i+1}), xi \in V(G)\,. \qquad (8)$$

If we consider the time consumed by a viewpoint plan, it may include:

(a) $n*t_1$ - time needed to acquire a view and transfer it to a 3D local model, including image digitalization, image preprocessing, 3D surface reconstruction, etc.

(b) $n*t_2$ - time for fusion and registration. Merge the local model with previous partial model.

(c) t_3 - time needed to perform the strategy of viewpoints planning. A plan of viewpoints and optical settings of the sensor are determined and a path is generated for the robot moving to the next pose.

(d) t_4 - time needed for the robot to perform the task of moving from one viewpoint to another.

Here t_3 is subject to the following constraints: ('≥' means the constraint condition is satisfied.)

	$g_1 \geq 0$	(resolution)
AND	$g_2 \geq 0$	(in-focus)
AND	$g_3 \geq 0$	(field of view)
AND	$g_4 \geq 0$	(visibility)
AND	$g_5 \geq 0$	(viewing angle)
AND	$g_6 \geq 0$	(overlap)
AND	$g_7 \geq 0$	(occlusion)
AND	$g_8 \geq 0$	(image quality)
AND	$g_9 \geq 0$	(kinematic reachability of sensor pose)

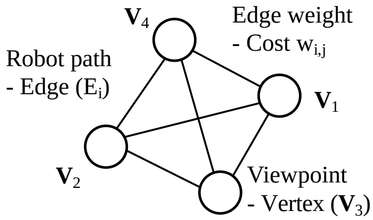

Fig. 2 A topology of viewpoint plan

If n viewpoints of image acquisition are needed to finish the task, the total needed time is [$(t1+t2)*n+t3+t4$].

Since there exists a priori model of the object, the planning strategy may run offline and t_3 is eliminated from the above equation. Assuming that t_1 and t_2 are constants and t_4 is proportional to the path length, the task time becomes $T_{\cos t}=(T1+T2)*n+l_c k$.

It is obvious that reducing the number of viewpoints will improve the vision perception behavior. Therefore, the objective is to take lowest traveling cost T_{cost} through the planned viewpoints. In fact, if both the object model and the robot environment are specified, the length of shortest path of taking views is not varying very much and the traveling cost is just proportional to the number of viewpoints. Hence the objective becomes to minimize the number of viewpoints. An optimal solution of sensor placement contains the least number of viewpoints and the corresponding graph has a lowest order. This is determined by HGA in the next section.

3. Determination of Optimal Topology

3.1 HGA Representation

Hierarchical GA is used to determine the optimal topology of sensor placements which contain minimal viewpoints with highest accuracy and satisfy all possible constraints. The hierarchical chromosome can be regarded as the DNA that consists of the parametric genes. In this paper, parametric genes (V_i) mean the sensor poses and optical settings and control genes (c_i) mean the topology of viewpoints. To indicate the activation of the control gene, an integer "1" is assigned for each control gene that is being enabled where "0" is for turning off. When "1" is signaled, the associated parameter genes due to that particular active control gene are activated in the lower level structure. However, the inactive genes always exist within the chromosome even when "0" appears.

For the sensor placement problem, a chromosome in GA represents a group of viewpoints with specific topology. $V_i=(x,y,z,a,b,g,a,f,d)$ represents a variable viewpoint, where $x,y,z \in R$, $a,b,g \in [-p,p]$, $a \in [a_{\min},a_{\max}]$, $f \in [f_{\min},f_{\max}]$, and $d \in [d_{\min},d_{\max}]$, and the corresponding $c_i=\{0,1\}$ represents a control gene which is a binary variable.

3.2 Plan Evaluation

In this paper, a plan of sensor placements is evaluated by a min-max criterion, which includes three objectives and a fitness evaluation formula.

The order of a graph ***G*** is equivalent to the number of occurrences of "1" in the control level genes. To plan a group of viewpoints with minimum order of the topology, the first objective is represented as:

Objective 1: minimize $o(G) = \sum_{i=1}^{n\max} c_i$. (9)

Assume that the accuracy of vision inspection is proportional to the surface resolution acquired by vision sensor and consider there are m features to be acquired, so the second objective is to improve the average accuracy, i.e.

Objective 2: maximize $h(F) = \frac{1}{m}\sum_{j=1}^{m} \frac{w_{j,image}}{l_j}$ (10)

where w_{image} is the length of a feature on the sensor image.

On the other hand, an admissible viewpoint is subject to 9 constraints in sensor placement space, i.e. resolution, in-focus, field of view, visibility, viewing angle, overlap, occlusion, contrast, and reachability. We set up a penalty scheme to handle these constraints such that invalid chromosomes become low performers in the population. The constrained problem is then transformed to an unconstrained condition by associating the penalty with all the constraint violations. We use a vector of penalty coefficients to combine the nine constraints:

K = $(d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9)$. (11)

Define a binary function

$$j_i = \begin{cases} 0, & \text{the constraint is satisfied} \\ 1, & \text{the constraint is violated} \end{cases} \quad (12)$$

and construct another vector of constraints:

Q(***l***, ***V***)= $(j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8, j_9)$, (13)

where ***l*** is an object feature and ***V*** is a viewpoint.

Therefore the third objective is to minimize the total penalties for the constraints:

Objective 3: minimize $penalty = K \cdot Q^T$. (14)

If there are m features and n viewpoints, the average penalty of a viewpoint topology is:

$$penalty = \frac{1}{m}\sum_{i=1}^{m} \min[K \cdot Q(l_i, V_j)^T \mid j = 1,2,...,n] . \quad (15)$$

Finally, we comprise the penalty scheme with the two objective functions and the fitness function is derived:

Fitness*: f(**G**)=* $(a \cdot n\max + b \cdot \ell_{\max} + |K|)$

$$- a \times o(G) - \frac{b}{h(F)} - K \cdot Q^T , \quad (16)$$

where $|K| = \sum_{i=1}^{m} K_i$, $(a \cdot n\max + b \cdot \ell_{\max} + |K|)$ is the maximum possible value that ensures positive fitness, $\ell_{\max}$ is maximum possible resolution, and a and b are two scaling factors.

3.3 Evolutionary Computing

According to the characteristics of sensor placement problem, the following genetic parameters and operations are suggested:

(a) Chromosome length:
 2*n*, where *n* stands for maximum viewpoints,

(b) Crossover method:
 control level genes: one-point crossover if n<10, two-point crossover if n>=10; probability of crossover $p_c = 0.25$;
 parametric level genes: Heuristic crossover with ratio=0.8. (Because the parameters of sensor pose and optical settings are real numbers.)

(c) Mutation method:
 control level genes: bit-flap mutation; probability of mutation $p_m = 0.01$
 parametric level genes: $g = g + f(m, s)$ where f is Gaussian distributed function, m and s are the mean and variance, respectively.

(d) Selection method: Roulette-Wheel selection method;

(e) Replacement: Steady State without duplicates;

(f) Population size: 30-100, based on the length of chromosome;

(g) Initial population: randomly generated.

4. Determination of a Shortest Path

For giving an efficient plan, a shortest path must be found through the above-determined optimal viewpoints. Define the edge weight connecting two vertices in the sensor placement graph to be $w(V_i, V_j)$, which is a nearest distance for the robot to move the sensor from point V_i to V_j.

Obviously a sensor placement graph satisfies the triangle inequality, i.e.

$w(V_i, V_j) \le w(V_i, V_k) + w(V_k, V_j), \forall V_k \in V(G) \setminus \{V_i, V_j\}$,

where the "=" comes into truth if the position of $\boldsymbol{V}_k$ is on the path l_{ij} and the orientation of $\boldsymbol{V}_k$ is a middle angle between Ω_i and Ω_j.

Here assumes that the robot should resume to its initial state after finishing the vision task. Given a specified graph, now another fundamental task is to find an optimal closed chain that is the shortest (or approximately shortest) one of all possible chains.

Because a sensor placement graph ***G*** is a finite, connected, and complete, the optimal closed chain is the optimal Hamilton cycle. Furthermore a complete graph must contain Hamilton cycles, i.e. there exists cycles which contain all the vertices once. In graph theory, it has been proved that if $\boldsymbol{G}$ is complete and satisfies triangle inequality, the optimal chain $\boldsymbol{C''}$ in a connected and weighted graph $\boldsymbol{G''}$ is corresponding to an optimal cycle $\boldsymbol{C}$ in its complete and weighted graph $\boldsymbol{G}$. That is, $C'' \leftrightarrow C \quad and \quad w(C'') = w(C)$, where $w(X)$ means the length of chain or cycle X.

To plan a sequence of robot operations or to find an optimal Hamilton cycle, we have to decompose $\boldsymbol{G}_n$ into the union of some edge-disjoint Hamilton cycles. There are total n vertices and $\partial(G) = \frac{1}{2} n(n-1)$ edges in the graph $\boldsymbol{G}_n$, and a Hamilton cycle $\boldsymbol{C}$ must contains n edges too. Let a Hamilton cycle be a sequence of vertices: $\boldsymbol{C}=(x_1, x_2, \ldots, x_n)$ where $xi \neq xj, xi \in V(G), i \in [1, n]$. The problem might be solved by enumerating all possible Hamilton cycles $\boldsymbol{C}_i$ in the graph, by comparing their sum weighs $w(\boldsymbol{C}_i)$, and then finding out the smallest one $cost=\min[w(\boldsymbol{C}_i)]$. However, there are total $o(C) = \frac{1}{2}(n-1)!$ Hamilton cycles. When n is a large number, it will bring unacceptable computations. e.g. $o(C) = 6 \times 10^{16}$ when n=20. It is a non-deterministic polynomial complete (NPC) problem in graph theory and must be solved by approximation algorithm.

This paper uses an approximation algorithm developed by Christofides. The procedures of this algorithm for finding an optimal Hamilton cycle is described as:

(a) Construct the distance matrix **W** from graph ($\boldsymbol{G}$, w).
(b) Find the smallest tree **T** in **W** using *Prim algorithm*
(c) Find the odd degree set **V** in **T** and calculate the *perfect matching* **M** of smallest weighs in $\boldsymbol{G'}=\boldsymbol{G}[\boldsymbol{V}]$ using *Edmonds-Johnson algorithm*.
(d) Find an Euler circuit $\boldsymbol{C}_0=(x1, x2, x3, \ldots, xn)$ in $\boldsymbol{G}^*=\mathbf{T}+\mathbf{M}$ using *Fleury algorithm*.
(e) Start at $x1$ and go along $\boldsymbol{C}_0$, remove each multi-occurrence vertex from $\boldsymbol{C}_0$ except for the last $x1$ and finally form a Hamilton cycle $\boldsymbol{C}$ of graph $\boldsymbol{G}$ and it is just the approximated optimal cycle.

The resulted Hamilton cycle is an approximation solution. It has been proven that the error ratio doesn't exceed 0.5 even in worst case. That is, if L_0 is the optimal solution (sum of weighs) and L is the approximation solution by Christofides algorithm, i.e. $1 \le L/L_0 \le 1.5$.

5. Experiments

5.1 Task Specification and System Setup

In this paper the experiments are carried out by computer simulations. Fig. 3 shows a CAD model used for examples of 3-D inspection. The part size is 300mm × 150mm × 180mm. There are total 14 surfaces. The vision task is to ensure full visibility of the part (13 surfaces) except for the bottom surface because it is on the conveyor. Constrained viewing angle is ±45° and resolution is 0.1mm/pixel. Assume there is no overlap constraint and a viewpoint outside the object is considered reachable to the robot.

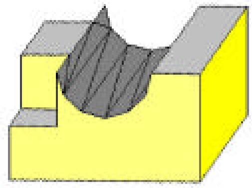

Fig. 3 The model of the object to be inspected

The essential components of the vision system are: a pair of CCD-cameras with the image resolution of 1024 by 1024, a 50mm lens, an one-point light source, a PC for image processing and HGA evolutionary computing, suitable software, a monitor to show the worked up images, a robot hand-eye system for moving the object around the object, a controller that executes the commands output from the computer, etc.

The vision sensor is a parallel stereo pair with the baseline about b=200mm. Assume the optical parameters (f, a) of the sensor are fixed but d can be adjusted dynamically. So there are total three fixed parameters (b, f, a) of the stereo pair and the corresponding values of these parameters are calibrated firstly.

5.2 Computing the Optimal Placement Topology

With the specified object model, the optimal solution of sensor placements was determined off-line using HGA. In our experiments, the maximum viewpoints was set to n_{max}=120. Beside some optical parameters of vision sensor are not variable, the self-rotation angle w of stereo pair is also not considered. So there are only two parameters related to the sensor orientation, i.e. q and j based on sphere coordinate system and the orientation can be expressed as a unit direction vector:

$\vec{P}_d = (x\vec{\mathbf{i}}, y\vec{\mathbf{j}}, z\vec{\mathbf{k}})$, where

$x = \cos j \cos q$, $y = \cos j \sin q$, and $z = \sin j$.

The dimension of a viewpoint vector is reduced to be 6-dimensional V_i=(x, y, z, q, j, d) where $q\,\hat{I}$ [-π, π], $j\,\hat{I}$ [-π/2, π/2] are entries of unit direction vector, in which each entry is represented as a 4-byte float value. The population size is set to 50. All other genetic parameters and operations are adopted using the default settings introduced in section 3.3.

The evolutionary computing was then carried out and the complete condition is based on fitness stability. Usually after hundreds of thousands of generations, a solution was obtained with order n=33 and some of the viewpoints are illustrated in Table 1.

5.3 Computing a Shortest Path

We first need to compute the distance matrix $\boldsymbol{D}_m=\{d_{ij}\}$ where d_{ij} is the distance between point V_i and V_j. The result is illustrated in Table 2. An approximation optimal Hamilton circle was determined by Christofides algorithm:

$L_{shortest}$= 75040, through ($\boldsymbol{V}_{25}$, $\boldsymbol{V}_{27}$, $\boldsymbol{V}_{14}$, $\boldsymbol{V}_{17}$, $\boldsymbol{V}_{11}$, $\boldsymbol{V}_{8}$, $\boldsymbol{V}_{10}$, $\boldsymbol{V}_{15}$, $\boldsymbol{V}_{31}$, $\boldsymbol{V}_{16}$, $\boldsymbol{V}_{21}$, $\boldsymbol{V}_{3}$, $\boldsymbol{V}_{9}$, $\boldsymbol{V}_{18}$, $\boldsymbol{V}_{1}$, $\boldsymbol{V}_{29}$, $\boldsymbol{V}_{28}$, $\boldsymbol{V}_{22}$, $\boldsymbol{V}_{26}$, $\boldsymbol{V}_{23}$, $\boldsymbol{V}_{7}$, $\boldsymbol{V}_{32}$, $\boldsymbol{V}_{5}$, $\boldsymbol{V}_{4}$, $\boldsymbol{V}_{2}$, $\boldsymbol{V}_{20}$, $\boldsymbol{V}_{13}$, $\boldsymbol{V}_{12}$, $\boldsymbol{V}_{30}$, $\boldsymbol{V}_{6}$, $\boldsymbol{V}_{33}$, $\boldsymbol{V}_{24}$, $\boldsymbol{V}_{19}$, $\boldsymbol{V}_{25}$).

Table 1. Some of the 33 viewpoints planed

	x	y	z	*q*	*j*	*d*
$\boldsymbol{V}_1$	-90.7	-271.5	426.7	0.85	-0.87	58.33
$\boldsymbol{V}_2$	438.2	87.0	1307.9	-2.85	-1.34	52.31
$\boldsymbol{V}_3$	640.9	-27.4	402.8	3.09	-0.69	55.84
$\boldsymbol{V}_4$	804.3	980.0	313.2	-2.16	-0.26	52.74
$\boldsymbol{V}_5$	1058.6	759.0	800.1	-2.45	-0.59	52.78
$\boldsymbol{V}_6$	-903.8	616.4	597.3	-0.53	-0.46	52.37
$\boldsymbol{V}_7$	1047.0	128.4	565.4	-3.00	-0.56	53.69
$\boldsymbol{V}_8$	-249.8	-1348	115.0	1.28	-0.08	52.18
$\boldsymbol{V}_9$	399.3	-276.4	428.0	2.30	-0.85	57.10
…	…	…	…	…	…	…
$\boldsymbol{V}_{32}$	1116.0	92.2	45.8	-3.05	-0.05	53.25
$\boldsymbol{V}_{33}$	232.4	-209.4	1307.9	1.95	-1.40	52.71

Table 2. Distance matrix of the viewpoints

	$\boldsymbol{V}_1$	$\boldsymbol{V}_2$	$\boldsymbol{V}_3$	$\boldsymbol{V}_4$	$\boldsymbol{V}_5$	$\boldsymbol{V}_6$	…	$\boldsymbol{V}_{27}$	$\boldsymbol{V}_{28}$	$\boldsymbol{V}_{29}$	$\boldsymbol{V}_{30}$	$\boldsymbol{V}_{31}$	$\boldsymbol{V}_{32}$	$\boldsymbol{V}_{33}$
$\boldsymbol{V}_1$	0	8167	4128	7919	8052	7349		4459	3223	2496	4743	8059	7773	6694
$\boldsymbol{V}_2$	8167	0	7869	2741	2015	4093		4425	5573	6129	5200	7712	3043	5172
$\boldsymbol{V}_3$	4128	7869	0	7132	7299	6701		3881	4857	4747	4753	4506	7295	4416
$\boldsymbol{V}_4$	7919	2741	7132	0	1034	3453		4554	5027	5815	4291	6575	2021	5913
$\boldsymbol{V}_5$	8052	2015	7299	1034	0	3942		4568	5221	5918	4765	6810	1947	5836
$\boldsymbol{V}_6$	7349	4093	6701	3453	3942	0		3705	4887	5750	2921	6167	4855	4245
$\boldsymbol{V}_7$	7250	2554	6920	2222	1742	4814		4489	4608	5172	4667	6963	1203	6264
$\boldsymbol{V}_8$	7355	6311	5748	6056	6384	3986		4427	6509	6976	4076	3846	6446	3289
…														
$\boldsymbol{V}_{32}$	7773	3043	7295	2021	1947	4855		5106	5201	5820	4830	6591	0	6770
$\boldsymbol{V}_{33}$	6694	5172	4416	5913	5836	4245		2974	5729	6047	4385	3610	6770	0

6. Conclusions

In this paper, a plan of sensor placements is evaluated with three conditions, i.e. low order, high precision, and satisfaction of all constraints.

To achieve optimal topology of viewpoints, it is difficult to use conventional mathematical methods. As a numerical optimizer, HGA generates the solutions that are not mathematically oriented possesses an intrinsic flexibility and the freedom to choose desirable optima according to task specifications. In the case of that the order of the sensor placement graph is greater then 15, it becomes impossible to enumerate all possible robot paths for obtaining a shortest one. Hence, an approximation algorithm is used to determine the viewing sequence. Christofides algorithm is an effect one that can guarantee no more than 0.5 error even in worst case.

Compared with previous approaches, this paper provides a robust and complete solution for inspection tasks, including viewpoint decision and robot operation sequence planning.

Acknowledgement: The work described in this paper is supported by a grant from the Research Grants Council of Hong Kong [Project No. CityU1136/98E].

References

[1] E. Marchand, F. Chaumette, "Active sensor placement for complete scene reconstruction and exploration", IEEE Int. Conf. on Robotics and Automation, 1997, pp. 743-750.

[2] H. Zha, K. Morooka, T. Masegawa, "Next best viewpoint (NBV) planning for active object modeling based on a learning-by-showing approach", Lecture Notes in Computer Science 1352, 1998, pp. II-185.

[3] J.E. Banta, M.A. Abidi, "Autonomous placement of a range sensor for acquisition of optimal 3-D models", Proc. of the 22nd Int'l Conf. on Ind. Elec., Contr., and Inst. Taipei, R.O.C., 1996, pp. 1583-1588.

[4] B. Triggs, C. Laugier, "Automatic camera placement for robot vision tasks", IEEE Int. Conf. on Robotics and Automation, 1995, pp. 1732-1737.

[5] J. Okamoto, M. Milanova, U. Bueker, "Active perception system for recognition of 3D objects in image sequences", 5th Int. Workshop on AMC, Coimbra, 1998, pp. 700-705.

[6] K.A. Tarabanis, R.Y. Tsai, P.K. Allen, "The MVP sensor planning system for robotic vision tasks", IEEE Trans.on RA-11:1, 1995, pp. 72-85.

[7] E.Trucco, M. Umasuthan, A.M. Wallace, V. Roberto, "Model-based planning of optimal sensor placements for inspection", IEEE Trans. on RA-13:2, 1997, pp. 182-194.

[8] Y. Ye, J. K. Tsotsos, "Where to look next in 3D object search", IEEE Int. Symp. for CV, Florida, 1995, pp.539-544.

[9] X.G. Gu, M.M. Marefat, F.W. Ciarallo, "A robust approach for sensor placement in automated vision dimensional inspection", Proc. IEEE Int. Conf. on Robotics and Automation, 1999, pp. 2602-2606.