NEAR-OPTIMAL-TIME PATH PLANNING IN CAD-GUIDED PART DIMENSIONAL INSPECTION [1]

Weihua Sheng [*] Ning Xi [*] Mumin Song [**] Yifan Chen [**]

[*] *Electrical and Computer Engineering Dept., Michigan State University, East Lansing, 48824, USA*
[**] *Scientific Research Laboratory, Ford Motor Company, Dearborn, MI 48121, USA*

Abstract: A CAD-guided vision sensor planning system is developed in our lab. To improve the efficiency of the inspection system, the path planning problem is rendered as a Traveling Salesman Problem(TSP). A new approach is developed to solve the TSP into its sub-optimality quickly. First, viewpoints are clustered into several groups. Second, clustered Traveling Salesman Problem is solved by favoring the inter-group paths. Experimental results on real parts show that the proposed approach is effective.

Keywords: CAD, Path Planning, Traveling Salesman Problem

1. INTRODUCTION

1.1 *CAD-Guided Vision Sensor Planning*

Sensor planning (Tarabanis *et al.*, 1995), or finding the suitable sensor configurations so that an inspection task can be carried out satisfactorily, is essential to automated inspection systems. A camera positioning system is being developed in our lab to plan and realize the camera configurations in a fully automated, accurate and efficient way(W.Sheng *et al.*, 2000). The overall system is shown in Figure 1. This system can be used to aid the automotive part dimensional inspection using structured light method. Based on the requirements of the structured light method, the problem of camera planning for dimensional inspection can be stated as follows:

find a set of camera viewpoints (position and orientation) to cover all the surfaces of the part and, for any point and its small neighboring area on the surfaces of the part, at least one viewpoint exists such that the point and its neighboring area are visible, in the field of view, resolvable and in focus (W.Sheng *et al.*, 2000).

[1] Research partially supported under NSF Grant IIS-9796300, IIS-9796287, EIA-9911077 and DMI 0115355.

In this system, the camera planning algorithm consists of two main steps. First, the compound surfaces are decomposed into patches that satisfy a flatness constraint. Second, a combined viewpoint generation algorithm is proposed for each patch based on two existing sensor planning methods: generate-and-test and synthesis (Tarabanis *et al.*, 1995).

1.2 *Shortest Path Planning Problem*

After all the viewpoints are generated, the path planning problem then follows: find the minimum-time movement of the robot to carry out the inspection. Obviously, the time for a complete inspection of the whole part consists of 1) the time spent on the traverse among the viewpoints and 2) the time spent on the execution of inspection at all the viewpoints. Since the latter is constant, the optimization of the total inspection time is, essentially, the minimization of the time spent on the traverse of the camera among viewpoints. Mainly, two factors determine the traverse time: 1) the trajectory, or time history of joint positions, velocities, accelerations, and torques, between each pair of viewpoints; 2) the order to visit all the viewpoints. In this paper, we focus on the latter.

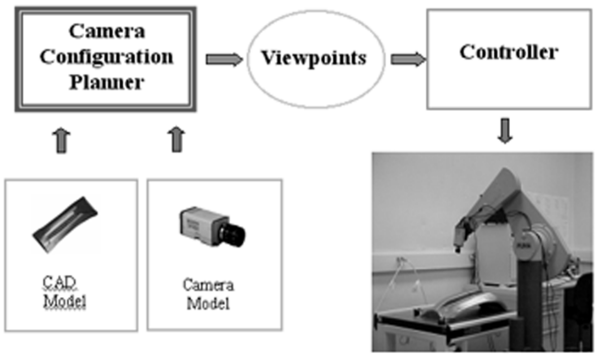

Fig. 1. Camera positioning system for automated part dimensional inspection

For a robotic system, the minimum-time geometric path of the robot's end-effector is not necessarily equivalent to the path of minimum Euclidean distance, but depends on the manipulator's kinematics and its inertial parameters (K.G.Shin and N.D.Mckay, 1986). However, in our view, Euclidean distance is still a good approximation.

Edan *et al.* developed a computationally efficient algorithm that allows the determination of the near-minimum-time path between n task points (Y.Edan *et al.*, 1991). They modeled the problem as a Traveling Salesman Problem (TSP) (E.L.Lawler *et al.*, 1985). To reduce the computation time, they divided the task points into several groups and solved the TSP group by group. However, they did not talk about how to find the paths connecting different groups, which will be an additional problem as the number of groups increases.

To model the robot path planning problem, a graph, $G = (V, E)$ is constructed in the following way: take the position of each viewpoint as a vertex v_i, the path between two vertices v_i and v_j as an edge e_{ij} and the minimum distance to traverse between these two vertices as its weight, w_{ij}. The goal is to find a shortest Hamiltonian path, or a path that visits all the vertices once and does not go back to the starting vertex. The following assumptions hold: 1) the Euclidean distance between two viewpoints corresponding to vertex v_i and v_j, $d(v_i, v_j)$, is taken as weight w_{ij}. 2) the graph is complete, or there exists a straight path between any pair of viewpoints.

1.3 *Traveling Salesman Problem and Its Variants*

The Traveling Salesman Problem tries to determine an order to visit n cities so that the total tour length is the shortest (E.L.Lawler *et al.*, 1985). In computation complexity theory, the TSP is NP-complete. There have been intensive research efforts in TSP. Basically, there are two types of algorithm for it. 1) exact algorithms and 2) approximation algorithms. The exact algorithms aim to find the optimal solutions, however, these algorithms are very slow, especially for large size problems. Some of the exact algorithms are: cutting planes, Branch-and-Bound and dynamic programming (E.L.Lawler *et al.*, 1985). The approximation algorithms try to find sub-optimal solutions using heuristic approaches and are much faster.

As variants of the TSPs, the shortest Hamiltonian path problems have three types:

- the shortest free-end Hamiltonian path (SFHP) problem – there is no specification on the end cities,
- the shortest one-end Hamiltonian path (SOHP) problem – one end city is specified, and
- the shortest two-end Hamiltonian path (STHP) problem – both end cities are specified.

All the three Hamiltonian path problems can be transformed into classical TSPs by enhancing the distance matrix (E.L.Lawler *et al.*, 1985). They can also be solved directly using approximation algorithms (N.Guttmann-Beck *et al.*, 2000; J.A.Hoogeveen, 1991).

The organization of the paper is as follows, in Section 2, the two-level-TSP approach is developed. Section 3 evaluates the performance of our approach. Section 4 provides the implementation and experimental results. Conclusions are provided in Section 5.

2. TWO-LEVEL-TSP APPROACH

It is apparent that the number of viewpoints will increase as the area of the part surfaces and the geometric complexity increase. Due to the nature of the TSP, the time to solve it grows rapidly as the problem size grows, even for approximation algorithms. On the other hand, based on the decomposition of compound surfaces, the resulting viewpoints tend to form groups in $3D$ space. That means the distances between viewpoints in the same group are relatively smaller than the distances between viewpoints in different groups. This geometric structure of the viewpoints inspires us to solve the problem in a hierarchical way:

cluster the viewpoints based on their distance similarity into several groups; solve the TSP for each group as well as determine the path to visit all the groups.

By taking advantage of the cluster nature of the problem, it is possible to find sub-optimal solutions very quickly. Following are some useful definitions when the hierarchical TSP is referred.

⋄ **inter-group edge**: an edge $e_{ij} = (v_i, v_j)$ with its vertices v_i and v_j in different groups.

⋄ **inter-group path**: the collection of inter-group edges that visit all the groups.

The shortest inter-group path is the shortest "connections" among all the groups. Apparently, if the number of the groups is N_g, the number of the inter-group edges on the inter-group path is $N_g - 1$.

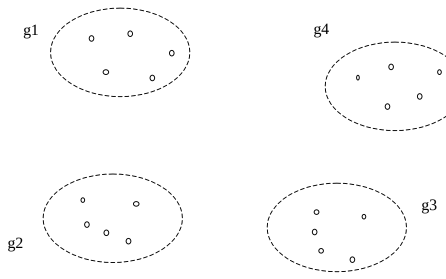

Fig. 2. An example problem

◇ **intra-group Hamiltonian path**: a path that visits all the vertices exactly once in a group without leaving the group.

Any overall path for N_g groups consists of one inter-group path and N_g intra-group Hamiltonian paths.

2.1 *Clustered Traveling Salesman Problem*

Once the viewpoints are clustered into groups, the two-level hierarchical TSP becomes the clustered Traveling Salesman Problem (CTSP) (J.A.Chisman, 1975):

Let $G = (V, E)$ be a complete graph with vertex set V and edge set E. The vertex set is partitioned into k groups, $g_1, g_2, ..., g_k$, determine a shortest path to visit all the vertices and the vertices of each group are visited consecutively.

To solve the CTSP, Chisman (J.A.Chisman, 1975) transformed the CTSP back to a TSP by adding big costs to the inter-group edges. However, his algorithm focuses on the group constraint instead of the computational cost. In this sense, his method does not reduce the computational cost of the original TSP. Lokin (F.C.J.Lokin, 1978) provided a Branch-and-Bound algorithm to exactly solve the CTSP, and as can be expected, it is not time-efficient. Approximation algorithms were also developed (M.Gendreau *et al.*, 1994).

Recently, Guttmann-Beck *et al.* (N.Guttmann-Beck *et al.*, 2000) proposed an approximation algorithm for CTSP. Their algorithms are based on a modified Christofides' algorithm (J.A.Hoogeveen, 1991) for the shortest Hamiltonian path in each group, as well as another modified Christofides' algorithm to find a shortest path connecting all the groups. Guttmann-Beck *et al.* determined the shortest intra-group Hamiltonian paths first, then calculated the paths among groups. Hence their algorithm favors the intra-group paths.

In this paper, a different perspective is taken to solve the CTSP, which favors the inter-group paths. Basically, two level of TSPs are solved. In the higher level, the shortest "connections", or the shortest inter-group path, among groups are determined first. In the lower level, with the entering and the leaving vertex already known in each group, the shortest Hamiltonian path problems are solved group by group. By combining the paths generated in both levels, a complete path can be obtained. As is analyzed later, this new algorithm has comparable performance with Guttmann-Beck's algorithm and in some cases it is even better.

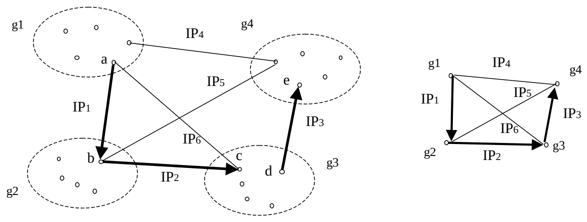

Fig. 3. The distances between groups and the group graph

2.2 *Level 1: Shortest Inter-group Path*

To find the shortest inter-group path, a new graph, called **group graph**, is constructed with each vertex representing a group and the distance between two vertices representing the distance between the corresponding two groups. Here the distance between two groups is defined as follows,

◇ **distance between two groups**: the minimum of the length of the inter-group edges between g_i and g_j. i.e., $D(g_i, g_j) = \min_{v_i \in g_i, v_j \in g_j} \; d(v_i, v_j)$.

An example problem is shown in Figure 2. The distances between groups and the corresponding group graph are shown in Figure 3, where $IP_1, IP_2, ...IP_6$ are the distances between each pair of groups.

The shortest free-end Hamiltonian path on the group graph is corresponding to the collection of the inter-group edges with minimum total distance that visit all the groups. In Figure 3, the thicker lines (IP_1, IP_2, IP_3) highlight the shortest inter-group path.

2.3 *Level 2: Shortest Intra-group Hamiltonian Paths*

As far as the way the shortest inter-group path enters and leaves each group is concerned, there are four possibilities as shown in Figure 3 :

1) in the starting group of the shortest inter-group path, there is only one leaving vertex,
2) in the ending group of the shortest inter-group path, there is only one entering vertex,
3) in some intermediate groups, entering and leaving vertices are different.
4) in some intermediate groups, entering and leaving vertices coincide.

Correspondingly, there are four ways to find the shortest intra-group Hamiltonian paths (or extended intra-group Hamiltonian paths as in the fourth case). Figure 4 shows them.

1) The shortest one-end Hamiltonian path (SOHP) problem is solved to find the shortest intra-group Hamiltonian path that ends at the starting vertex of the

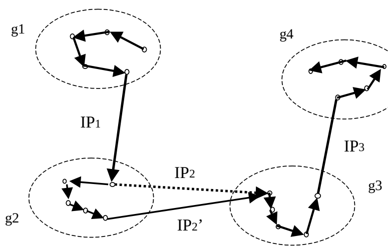

Fig. 4. Find the intra-group Hamiltonian paths

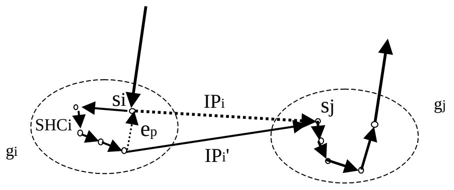

Fig. 5. The relation between the shortest Hamiltonian circle and the shortest extended intra-group Hamiltonian path

shortest inter-group path.
2) The shortest one-end Hamiltonian path (SOHP) problem is solved to find the shortest intra-group Hamiltonian path that starts at the ending vertex of the shortest inter-group path.
3) A shortest two-end Hamiltonian path (STHP) is found which begins at the entering vertex and ends at the leaving vertex.
4) A shortest two-end Hamiltonian path (STHP) is found which begins at the entering vertex of the current group and ends at the entering vertex of the next group. As a result, in the overall path, the corresponding inter-group edge is replaced by the new edge joining the two consecutive groups.

In the fourth case, the following lemma holds.

Lemma 2.1. Denote the length of the shortest two-end Hamiltonian path that extends from group g_i to the next group g_j as $d(STHP_i')$, the length of the shortest Hamiltonian circle in group g_i as $d(SHC_i)$, IP_i is on the shortest inter-group path and it joins g_i with g_j, denote $d(IP_i)$ as its length, then $d(STHP_i') \leq d(SHC_i) + d(IP_i)$.

Proof. In Figure 5, assume SHC_i is the shortest Hamiltonian circle in group g_i. Delete the edge (e_p) between the starting vertex s_i and the vertex preceding s_i on SHC_i. Connect the preceding vertex with the starting vertex s_j of the next group g_j by edge IP_i'. Thus an extended intra-group Hamiltonian path is constructed from s_i to s_j. Obviously, the length of this Hamiltonian path is no less than $d(STHP_i')$. On the other hand, due to the triangle inequality, the length of this Hamiltonian path is less than $d(SHC_i)$ + $d(IP_i)$. Therefore $d(STHP_i') \leq d(SHC_i) + d(IP_i)$. $\triangleleft$

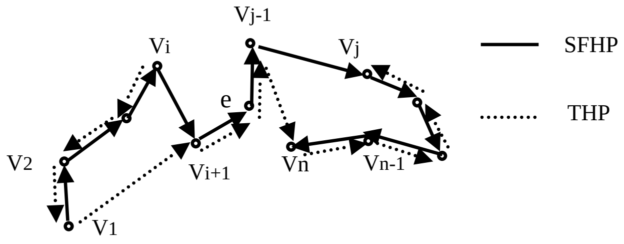

Fig. 6. Construct a two-end Hamiltonian Path from an SFHP

3. PERFORMANCE ANALYSIS

The performance of the two-level-TSP hierarchical algorithm is analyzed here. That is, what is the length difference between the solution of the TLT algorithm and the optimal solution of the CTSP? if both the shortest inter-group path and the shortest intra-group Hamiltonian path problems achieve optimality.

Lemma 3.1. Denote the total length of the shortest free-end Hamiltonian path of group g as $d(SFHP)$, the total length of the shortest two-end Hamiltonian path of the same group as $d(STHP)$, then the following inequality holds, $d(STHP) \leq d(SFHP) + 2D(g)$. Here $D(g)$ is the diameter of group g, or the maximum distance of all pairs of vertices in that group.

Proof. As shown in Figure 6, the n vertices are numbered sequentially from v_1 to v_n in the order of the shortest free-end Hamiltonian path ($SFHP$). Given any pair of starting and ending vertex, (v_i, v_j), the following method can construct a two-end Hamiltonian path.

1)v_i and v_j are adjacent on the $SFHP$. It is obvious that by adding an edge (v_1, v_n) and deleting an edge (v_{j-1}, v_j), a Hamiltonian path is obtained with given starting and ending vertices.
2)v_i and v_j are not adjacent. Similarly, by adding two edges, (v_1, v_{i+1}) and (v_{j-1}, v_n) and deleting two edges (v_i, v_{i+1}) and (v_{j-1}, v_j), a Hamiltonian path is obtained with given starting and ending vertices.

Hence, by adding at most two edges an $SFHP$ can be converted to a two-end Hamiltonian path with any given starting and ending vertices, which implies that the length of the $STHP$ of any pair of starting and ending vertices should be no more than the length of $SFHP$ plus 2 times of the diameter of the group, i.e., $d(STHP) \leq d(SFHP) + 2D(g)$. $\triangleleft$

Obviously, for each group, Guttmann-Beck's algorithm has a loss of $2D(g)$ due to favoring intra-group Hamiltonian paths. And the above lemma implies that the loss caused by favoring the inter-group path is also $2D(g)$. Hence they are comparable.

Similarly to lemma 3.1, we have the following lemma,

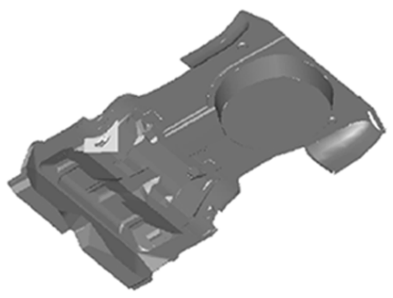

Fig. 7. A floor pan

Lemma 3.2. Denote the total length of the shortest free-end Hamiltonian path of group g as $d(SFHP)$, the total length of the shortest one-end Hamiltonian path of the same group as $d(SOHP)$, then the following inequality holds, $d(SOHP) \leq d(SFHP) + D(g)$.

Denote the total length of the optimal solution of the CTSP as $d(OPT)$. Obviously, the following inequality holds, $d(OPT) \geq d(IP) + d(SFHP)$. Here $d(IP) = \sum_{i=1}^{N_g-1} d(IP_i)$ is the total length of the shortest inter-group path solved based on the group graph. $d(SFHP) = \sum_{i=1}^{N_g} d(SFHP_i)$ is the sum of the length of the shortest free-end Hamiltonian paths in all groups. The equality holds only when in each group the two end vertices determined by the two-level-TSPs(TLT) algorithm are the same.

Theorem 3.1. Denote the total length of the overall path returned by the TLT algorithm as $d(TLT)$, then $d(TLT) \leq d(OPT) + 2\sum_{i=1}^{N_g} D(g_i)$.

Proof. The overall path returned by TLT algorithm consists of the inter-group path and the intra-group paths. For the starting and ending group where only one end vertex is given, as proved in lemma 3.2, $d(SOHP_i) \leq d(SFHP_i) + D(g_i)$. For the groups with two different entering and leaving vertices, lemma 3.1 has shown that $d(STHP_i) \leq d(SFHP_i) + 2D(g_i)$. For those shortest intra-group Hamiltonian paths that extend to the next group, as we have proved in lemma 2.1, $d(STHP_i') \leq d(SHC_i) + d(IP_i)$. While the length of the shortest Hamiltonian circle should be less than the length of the shortest free-end Hamiltonian path plus the diameter of the group, i.e., $d(SHC_i) \leq d(SFHP_i) + D(g_i)$, we have $d(STHP_i') \leq d(SFHP_i) + D(g_i) + d(IP_i)$.

Adding up all the paths we have

$$d(TLT) \leq \sum_{i=1}^{N_g} SFHP_i + 2\sum_{i=1}^{N_g} D(g_i) + \sum_{i=1}^{N_g-1} d(IP_i)$$

$$\leq d(SFHP) + d(IP) + 2\sum_{i=1}^{N_g} D(g_i)$$

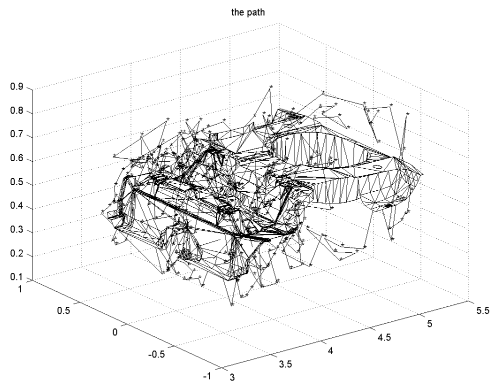

Fig. 8. The path by TLT algorithm

$$\leq d(OPT) + 2\sum_{i=1}^{N_g} D(g_i).$$

◁

It is obvious that the TLT algorithm obtains solutions within a constant bound from the optimal solution of CTSP. So the TLT algorithm is favorable for highly clustered CTSPs.

4. IMPLEMENTATION AND RESULTS

4.1 *Implementation*

The TLT algorithm is implemented in C. To cluster the viewpoints, a modified nearest neighbor algorithm is developed based on A.K.Jain's algorithm (A.K.Jain, 1988). This modified algorithm keeps the size of each group and the number of groups to be moderate. The shortest inter-group path, or the top level of the TSP is solved using a simulated annealing method based on a minimum spanning tree. For the SFHP, SOHP and STHP, a Christofides-like algorithm is used (J.A.Hoogeveen, 1991), which is based on the minimum spanning tree and the minimum matching.

4.2 *Results*

Parts from Ford Motor Company are used to test the algorithm. First, the viewpoints that satisfy the task constraints are generated. Second, these viewpoints are grouped into multiple groups. Third, using TLT algorithm, a complete path is obtained. As an example, a floor pan is used, which is shown in Figure 7. Totally 510 viewpoints are generated. The viewpoints are clustered into 46 groups. The overall path is shown in Figure 8, where the positions of the viewpoints are marked. The total length of the path is 59.96 m and the running time is 17.35 s on $Unix^{TM}$ with a $Sun\ Ultra$ 1 167 Mhz CPU and 512 M RAM .

We compare the TLT algorithm with SFHP algorithm that runs on the whole set of viewpoints (we call it one-level-TSP(OLT) algorithm). The overall path obtained by OLT is shown in Figure 9 with a path

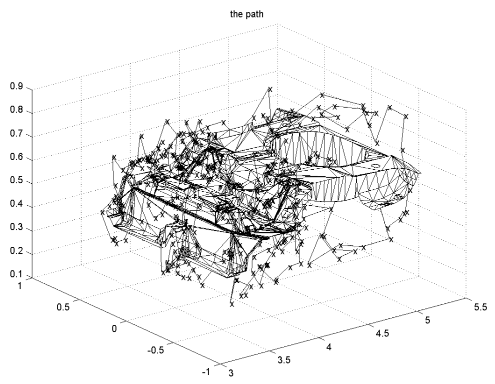

Fig. 9. The path by OLT algorithm

length of 45.26 m and running time 264.25 s. The following table summarizes the performance (path length and running time) of the TLT and the OLT algorithm on different parts.

	No. of Viewpoint	Length (m)		Time (s)	
		OLT	TLT	OLT	TLT
pillar	55	16.58	17.62	17.50	1.82
door	103	35.28	36.78	37.02	6.26
floorpan	510	45.26	59.96	264.25	17.35

So it is clear that the TLT algorithm is much faster than the one-level-TSP algorithm. The total lengths of the paths are comparable for both algorithms.

5. CONCLUSIONS

In this paper, how to efficiently solve the shortest path planning problem in CAD-guided vision sensor planning is discussed. A two-level-TSP (TLT) algorithm is developed based on variants of shortest Hamiltonian path problems. Performance of the TLT algorithm is analyzed. It is shown that a constant bound can be achieved which is related to the diameter of the clusters. This new algorithm can obtain near-optimal solutions quickly for many large scale TSPs, which are common problems in many robot path planning applications.

6. REFERENCES

A.K.Jain (1988). *Algorithms for Clustering Data*. Prentice Hall. New Jersey.

E.L.Lawler, J.K.Lenstra, A.H.G.R.Kan and D.B.Shmoys (1985). *The Traveling Salesman Problem*. John Wiley and Sons. Chichester.

F.C.J.Lokin (1978). Procedures for traveling salesman problems with additional constraints. *European Journal of Operational Research* (3), 135–141.

J.A.Chisman (1975). The clustered traveling salesman problem. *Computers and Operations Research* **2**, 115–119.

J.A.Hoogeveen (1991). Analysis of christofides' heuristic: Some paths are more difficult than cycles. *Operations Research Letters* **10**(5), 291–295.

K.G.Shin and N.D.Mckay (1986). Selection of near minimum time geometric paths for robotic manipulators. *IEEE Transactions on Automic Control* **31**(6), 501–511.

M.Gendreau, G.Laporte and J.Y.Potvin (1994). Heusitics for the clustered traveling salesman problem. Technical report. Centre de recherche sur les transports. Universite de Montreal.

N.Guttmann-Beck, R.Hassin, S.Khuller and B.Raghavachari (2000). Approximation algorihthms with bounded performance guarantees for the clustered traveling salesman problem. *Algorithmica* **28**, 422–437.

Tarabanis, K. A., P. K. Allen and R. Y. Tsai (1995). A survey of sensor planning in computer vision. *IEEE transaction on robotics and automation* **11**(1), 86–104.

W.Sheng, N.Xi, M.Song, Y.Chen and J.S. Rankin III (2000). Automated cad-guided automobile part dimensional inspection. In: *Proceedings of 2000 international conference on Robotics and Automation*. Vol. 2. pp. 1157–1162.

Y.Edan, T.Flash, U.M.Peiper, I.Shmulevich and Y.Sarig (1991). Near-minimum-time task planning for fruit-picking robots. *IEEE Transactions on Robotics and Automation* **7**(1), 48–56.