MULTIPLE ROBOT TASK DISTRIBUTION: TOWARDS AN AUTONOMOUS PHOTOGRAMMETRIC SYSTEM

GUSTAVO OLAGUE and ENRIQUE DUNN
Departamento de Ciencias de la Computación, División de Física Aplicada,
Centro de Investigación Científica y de Educación Superior de Ensenada, B.C.
Km. 107 Carretera Tijuana-Ensenada, 22860, Ensenada, B.C. México
golague@cicese.mx

Abstract

Automation of photogrammetric tasks by means of manipulator robots is a complex problem. It involves planning and controlling many aspects that reflect on the overall system performance in terms of precision and efficiency. This paper deals with the problem of task distribution for a multiple manipulator work cell with the goal of obtaining highly accurate object measurements. Task distribution is separated into two independent combinatorial optimization problems: activity assignment and tour planning. These problems are solved simultaneously by an optimization method based on genetic algorithms. This method implements a series of restriction-based heuristics in order to utilize a simple genetic representation similar to random keys. Experiments that validate the effectiveness of our approach are presented.

Keywords
Genetic algorithms, multiple robots, computational photogrammetry.

1 Introduction

Active vision is devoted to improving a perception task based generally on control strategies. Multiple parameters such as position and orientation of the sensors, focus or aperture, as well as the way data are processed; need elaborate control strategies in order to perform scene reconstruction or interpretation tasks. Several studies toward highly accurate reconstruction considering the above aspects can be found in the literature. Digital photogrammetry is one approach to solve such a problem. However, an automated robot vision system with the task of measuring three-dimensional objects needs to make decisions, which require a certain degree of intelligence. The problem we are interested in this paper is the task distribution among multiple robots, which we separate into two independent combinatorial problems: *activity assignment* and *tour planning*. The approach we follow is based on the evolutionary computation paradigm. The biologically computing paradigm is useful at solving, in a global way, the task distribution through a process of learning and adaptation.

Previous work of active vision systems involves a manipulator using a camera-in-hand configuration. The HEAVEN system developed by Sakane et al. (1992) is an example in which the camera and light illumination placement are studied. The MVP system developed by Tarabanis et al. (1993) considered the viewpoint planning of one manipulator monitoring the movements of a second robot. The work developed by Triggs and Laugier (1995) considers workspace constraints of a robot carrying a camera with the goal of automated inspection. Roberts and Marshall (1997) also used a single robot vision system for object recognition instead of accuracy tasks. More recently, Marchand and Chaumette (1999) studied optimal camera motion in active vision systems for 3D reconstruction and exploration. However, none of these systems have studied the problem of assigning and sequencing the best order of movements that a multiple robot system needs to perform.

2 Problem Statement

Photogrammetric network design is the process of placing cameras in order to perform photogrammetric tasks. The problem of configuring an optimal imaging geometry or network configuration is recognized as an open problem in photogrammetry. This problem is related to the viewpoint-planning problem in robotics. Multiple robots can be used to perform the photogrammetric project, e.g., measurement robots used in flexible manufacturing, see Figure 1. However, the problems of task distribution among robots and robot motion planning emerge. This paper is devoted to solving the task distribution problem.

In order to distribute the multiple viewpoints among the robots, the following questions arise:

1. How many and which viewpoints should be assigned to each robot?
2. What is the correct order of each viewpoint in a robot's tour?

Figure 1. Photogrammetric network simulation of four robots, each camera is mounted on the robot's hand, with the goal of measuring the box on the table.

The problem of task distribution then can be divided into two independent combinatorial problems: activity assignment and tour planning. In order to pose the problem, we give the following definitions:

1. The photogrammetric network is represented as an ordered set **V** of n three-dimensional viewpoints. Each individual viewpoint is expressed as Vj, where j ranges from $j=1$ to n.
2. An ordered set R consisting of r robots in the work cell represents a multi-robot active vision system. Each individual robot is expressed by Ri, where i ranges from $i=1$ to r.

From the above definitions, the activity assignment problem relates each of the n elements of **V** with one of the r possible elements of R, considering that each robot Ri has assigned n_i viewpoints, a problem of sequencing the viewpoints emerges which we call tour planning. Our goal is to find the best combination of activity assignment and tour planning in order to optimize the overall operational cost of the task distribution. This total operational cost is produced by adding individual tour costs, Qi, defined by the Euclidean distance that each robot needs to travel in straight lines among the different viewpoints. This criteria is represented as:

$$Q_T = \sum_{i=1}^{r} Q_i$$

Assuming there are n viewpoints to be assigned to r robots. The activity assignment problem presents r^n possible solutions. This means that the assignment problem search space grows exponentially with respect to the number of viewpoints. Additionally, the tour-planning problem still needs to be solved for each robot. A robot's tour consisting of n_i viewpoints will offer $n_i!$ sequencing alternatives. Therefore, the total number of sequencing alternatives for r robots is given by multiplying the number of alternatives of each robot. This is expressed by $\prod_{i=1}^{r} n_i!$. The scenario with the fewest possible sequencing alternatives is given by an equal distribution of viewpoints among the robots, where each robot is assigned $\frac{n}{r}$ viewpoints. Hence, the minimum number of possible tour planning possibilities for any given activity assignment, where n is a multiple of r, is given by $\left(\frac{n}{r}\right)!^r$. In this way, the total size of our task distribution search space, i.e., the amount of possible solutions to our problem, is greater than the product $r^n\left(\frac{n}{r}\right)!^r$.

The trivial solution to our combinatorial problem is to generate and evaluate each one of the possible solutions, carrying out an exhaustive search. This approach guarantees us to obtain the best possible solution. Unfortunately, it makes the problem computationally not tractable, since the computing time becomes prohibitive. The use of deterministic heuristics generally leads the search towards a single region of the search space, resulting in a local search. Hence, the quality of the solution depends on the validity of the heuristic's underlying assumptions. We need an approach that will yield a good solution in an acceptable computing time.

3 Our Approach

Our problem is presented as a combinatorial optimization problem with a large search space. An optimization method based on genetic algorithms is proposed. To obtain a quality solution three key aspects need to be addressed: search space reduction, solution representation and search heuristics.

3.1 Search space reduction.

Combinatorial problems generally have to satisfy a given set of competing restrictions. In our task distribution problem some of these restrictions are straightforward, i.e., each viewpoint should be assigned to only one robot, each viewpoint should be visited only once inside a robot's tour. On the other hand, implicit restrictions, like the accessibility of a robot to a particular viewpoint, need to be determined. Consideration of such restrictions can help reduce the size of the search space. The method by which such restrictions are computed is presented next. Assuming a static and obstacle free environment it is reasonable to compute the robots accessibility for a given position and orientation by means of solving the robot's inverse kinematics' problem. In this work we consider the PUMA560 manipulator, which consists of six degrees of freedom. A three-dimensional computer graphics simulation environment was developed in order to visualize such accessibility restrictions. Multiple manipulators were considered in our computer simulation.

The inverse kinematics' problem was solved for every robot at each viewpoint. The cases where a robot could access a viewpoint were stored in an auxiliary data structure called ACCESSIBILITY. This structure contains an entry for every viewpoint, Vj, in order to keep a record of how many and which robots are capable of reaching that particular viewpoint, see Table 1. Such values remain constant throughout the course of task execution; therefore they need only be computed once.

Viewpoint ID	Number of Robots	List of Robots ID's
V_1	r_1	$RobID_1,\ldots,RobIDr_1$
:	:	:
V_n	r_n	$RobID_1,\ldots,RobIDr_n$

Table 1. Structure ACCESSIBILITY, contains the number and the list of robots capable of reaching a particular viewpoint.

The above method evaluates the restrictions imposed by the physical arrangement of the work cell, as well as the robot's revolute joint limitations. Such operational restrictions are incorporated implicitly as an intrinsic element of our optimization method.

3.2 Solution Representation.

A representation similar to random keys (Bean 1994) is proposed. In this representation, each viewpoint *Vj* is assigned a random value *Sj* in the range (0, 1). These values are stored in a representation string denoted by **S**. Since there are *n* different viewpoints, **S** will consists of *n* elements.

Random keys use a heuristic we call the smallest-value-first heuristic. In our case, the viewpoint with the smallest corresponding value in **S** would be the first viewpoint in a given permutation **P**. The viewpoint with the second smallest value in **S** would be the second viewpoint in **P**, and so forth. In this way, the order of a viewpoint *Vj* inside a given permutation **P** depends on the magnitude of its corresponding value, *Sj*, with respect to all the other values in **S**. To illustrate, given five viewpoints a possible representation string can be

S=[0.89 0.76 0.54 0.23 0.62].

The smallest value in **S** can be found at the fourth position, denoted by S_4. Therefore, V_4 is the first viewpoint in the resulting permutation **P**. The second smallest value is found in the third position S_3, making V_3 the second viewpoint in **P**, and so on. The resulting permutation of the five viewpoints is

$\mathbf{P}=[V_4, V_3, V_5, V_2, V_1]$.

The random keys approach can be adapted to solve our task distribution problem. The smallest-value-first heuristic, avoids the generation of infeasible solutions common to permutation-based representations. Random keys representation also allows our optimization method to apply genetic operators without the need for additional heuristics. However, to accommodate this representation to our activity assignment, additional heuristics need to be incorporated.

Robot ID	Number of Viewpoints	List of Viewpoint ID's
R_1	v_1	$T1=[ViewID1,\ldots,ViewIDv_1]$
:	:	:
R_r	v_r	$Tr=[ViewID1,\ldots,ViewIDv_r]$

Table 2. Structure TASKS, contains the list of viewpoints comprising each robot's tour, ***Ti****.*

The convention of encoding a possible solution into a string representation has been specified. The question of how to describe the corresponding solution to such a representation is now considered. Recalling the problem statement, initially there is a set of *n* viewpoints *Vj*, and each must be assigned to one of the *r* possible robots. Using random keys representation, a possible solution is codified into a string S of *n* values. As stated in Section 2, we want to optimize the total operational cost Q_T. However, the solution representation S needs to be decoded into an explicit description of the task distribution problem. Such a description would represent each of the *r* robot tours. To accomplish this, an auxiliary data structure called TASKS is proposed to represent global task distribution among robots, see Table 2. This structure has an entry Ti for each robot *R*i, which describes that robot's tour, i.e., Ti lists the sequence of viewpoints assigned to that particular robot. Each of these tours Ti is evaluated to obtain individual tour cost *Q*i, from which the total operational cost Q_T is obtained. The question before us now is: how to convert a string representation into a corresponding task distribution description. The following subsection presents the heuristics used by our method to obtain such task distribution description.

3.3 Search Heuristics.

A solution representation **S**, needs to be evaluated. Such evaluation is applied to the task distribution description contained in TASKS. Hence, a mapping M: S→TASKS is necessary. M assigns and sequences the viewpoints among the different robots and stores the results in the structure TASKS. The mapping M makes use of the solution representation data structures **S** and TASKS, as well as the precomputed operational restrictions stored in ACCESSIBILITY. The activity assignment problem allocates each of the viewpoints *Vj* to one of the possible robots. The goal is to provide an initial unsequenced set of individual robot's tours ***Ti***, using the following steps:

1 Obtain the r_j number of robots capable of reaching that particular viewpoint by consulting the ACCESSIBILITY structure, see Table 1.
2 Divide the interval (0,1) in r_j equally distributed segments; in order to determine the size of a comparison segment $Seg = \frac{1}{r_j}$.
3 Calculate in which *k* segment the random value *Sj* resides, that is $k = Int\left(\frac{Sj}{Seg}\right) + 1$.
4 Assign the viewpoint Vj to the k-*th* robot in the corresponding entry in the ACCESSIBILITY table. In this way, the assigned robot index *i* is given by RobIDk, which is found on the entry in ACCESSIBILITY which corresponds to *Vj*.
5 Append Vj to the list of viewpoints, ***Ti***, assigned to i-*th* robot. This tour description ***Ti*** is stored in the TASKS structure.

The above heuristic's steps are based on the compliance with operational restrictions, and in doing so; assure that any codified string **S** brings a valid solution to the assignment problem. Based on such strategy, each possible codification string **S** has only one possible interpretation. After this series of steps each viewpoint is assigned to a robot. The viewpoints assigned to a single robot *Ri*, can be grouped into a set ***Ti***. Each ***Ti*** represents a tour of viewpoints assigned to that particular robot. These tours are stored in the structure TASKS. However the order of each viewpoint inside a given tour has not been specified. This is the problem we approach next. The tour planning problem consists of correctly sequencing each of the *r* robot tours ***Ti*** stored in the structure TASKS. These tours are initially obtained from the activity assignment phase presented above, in which every viewpoint *Vj* is assigned to one of the *r* possible robots, ***Ri***. The goal of the tour-planning phase is to minimize the total operational cost Q_T. This situation is equivalent to solving r different traveling salesman problems.

The smallest-value-first heuristic can be applied to sequencing problems such as the one presented here. Unfortunately, the rules by which the preceding assignments were made in steps 1 through 4, produces undesirable tendencies in the representation values *Sj* that correspond to each tour specification ***Ti***. This is due to the deterministic heuristic applied for robot assignment. As a consequence, the values corresponding to the viewpoints contained in ***Ti*** will be, on the average, higher than those corresponding to the viewpoints in T_{i-1} and will create a bias inside each ***Ti*** when directly applying the smallest-value-first heuristic. Therefore, the values inside **S** need to be adjusted to eliminate such unwanted properties. The following heuristic's steps accomplish this:

6 Recall in which of the *k* possible segments of the range (0,1) lies the S_j value used in the assignment phase.
7 Calculate the value S'_j in the range (0,1) that reflects the relative position of S_j inside the k-*th* segment. For example: Consider the value 0.70, which lies inside the range (0.60, 0.80), this value lies exactly in the middle, hence its corresponding value in the range (0,1) is 0.5.
8 Update S_j to store the new value S'_j.
9 Apply the smallest-value-first heuristic to each of the unordered robot tours ***Ti*** using the values stored in **S**.

After these adjustments have been made, we can proceed to apply the smallest-value-first heuristic to each tour ***Ti*** using the new values stored in **S** and rearrange each tour accordingly.

4 Experimentation and Results

The solution presented in this work for the task distribution problem was incorporated as an extension of the functionality of the EPOCA system developed previously (Olague and Mohr, 2001). EPOCA solves the photogrammetric network design problem for complex objects. The problem of task distribution emerges as a result of the photogrammetric network design performed by EPOCA. The system can be classified as an evolutionary computation based system that addresses the complex goal of automating the planning of sensing strategies for accurate three-dimensional reconstruction. Two different experiments are presented next: the first is a simple scenario intended to illustrate our method's functionality, the second experiment is somewhat more complex and it's goal is to show the effectiveness and flexibility of our system.

3.3 Experiment A

This experiment consists of eight viewpoints to be distributed among four manipulators. The viewpoints are stacked into four pairs, each pair arranged beneath one of the robots initial position. The optimal task distribution for this example can be obtained using a greedy heuristic. Hence, such an experiment might seem trivial, but it will exemplify our method's functionality. Operational restrictions are computed first, with the goal of determining which robots can access a particular viewpoint. As mentioned in section 3, to compute such restrictions the inverse kinematics' problem is solved for every robot at each viewpoint. The results of such validations are stored in the structure ACCESSIBILITY. The physical arrangement of the robots for experiment A is such, that every camera can be reached by three different robots, see Table 3.

Viewpoint ID	Number of Robots	List of Robots ID's
V_1	$r_1=3$	R1 R2 R3
V_2	$r_2=3$	R1 R2 R3
V_3	$r_3=3$	R1 R3 R4
V_4	$r_4=3$	R1 R3 R4
V_5	$r_5=3$	R1 R2 R4
V_6	$r_6=3$	R1 R2 R4
V_7	$r_7=3$	R2 R3 R4
V_8	$r_8=3$	R2 R3 R4

Table 3. ACCESIBILITY restrictions calculated for experiment A.

The genetic algorithm works with a population of codified strings, selecting the best individuals for reproduction. Such reproduction process combines the characteristics of two selected *parent* solutions and provides two new *offspring* solutions, which, in turn, will be part of the next *generation* of solutions. This process is repeated in an iterative manner until a certain number of generations are executed. At the end of this iterative process we obtain a set of possible solutions. One of those individuals, who represented the optimal solution, was given by the following random keys representation:

S=[0.72, 0.71, 0.32, 0.14, 0.81, 0.80, 0.27, 0.07].

After the assignment heuristic, we determine in which of the k segments each element Sj resides. For the first viewpoint, V_1, there are 3 possible robots to be assigned (see Table 3), hence the comparison segment Seg = 0.33. In this way, following steps 1 trough 5, the corresponding representation value S_1=0.72 is determined to be in the third segment, which is delimited by (0.66, 1.00). Therefore, the robot to be assigned is the third robot on V_1's entry on the structure ACCESSIBILITY, in this case RobID=3. The corresponding robot to be assigned to each viewpoint Vj is given as follows:

Robot =[R3 R3 R1 R 1 R4 R4 R2 R2].

At this point we have an appropriately assigned set of viewpoints. The values contained in S will now be adjusted in accordance with steps 5 through 9 so that the smallest-value-first heuristic can be applied to the viewpoints assigned to each robot. For the first viewpoint it's corresponding value S_1 is adjusted as follows. Recall that S_1=0.72 resides on the third segment which is delimited by (0.66, 1.00). The corresponding value of 0.72 on the range (0, 1) with respect to the third segment just mentioned is given by the value 0.18. Applying these steps to every value in **S** yields: **S**=[0.18, 0.15, 0.96, 0.42, 0.45, 0.42, 0.81, 0.21]. Once the values in S have been adjusted, applying the *smallest-value-first* heuristic rearranges TASKS as shown in Table 4.

Robot ID	Number of Viewpoints	List of Viewpoints ID's
1	2	$\mathbf{T}_1=[V_4\ V_3]$
2	2	$\mathbf{T}_2=[V_8\ V_7]$
3	2	$\mathbf{T}_3=[V_2\ V_1]$
4	2	$\mathbf{T}_4=[V_6\ V_5]$

Table 4. TASKS represent an optimal solution for Experiment A after the tour-planning phase.

3.4 Experiment B

This experiment presents a complex planar object, which is measured by four manipulators. The goal is to distribute the photogrammetric network consisting of 13 cameras in an optimal manner, see Figure 2. Working with this fixed configuration we executed several tests. First, to test our method's functionality we executed the task distribution planner. Several possible solutions are obtained over the course of multiple executions; one of such solutions is depicted in Figure 2. Notice that the best solution found does not incorporate all of the available robots. Furthermore, to test the method's adaptability two of the four-manipulator robots were disabled. The system is expected to distribute tasks between the two remaining robots. Results from such tests are shown in Figure 3. In these cases the *activity assignment* problem becomes visually simpler to resolve but the difficulty of the *tour-planning* problem becomes more evident since each tour will consist of more viewpoints.

Figure 2. Best solution found by the genetic algorithm. 13 viewpoints are to be distributed among Four manipulators. Viewpoints are depicted as individual cameras.

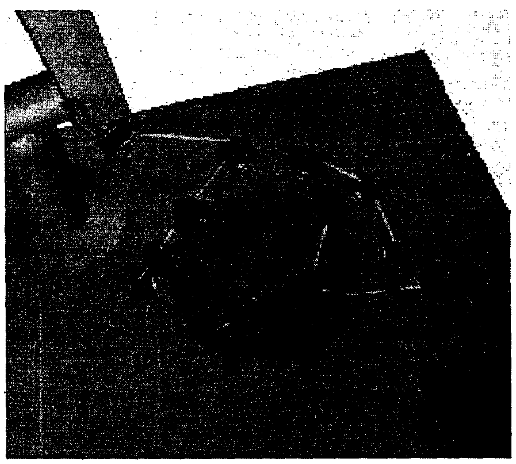

Figure 3. Solution found by the system for the case where a pair of robots are disable from the configuration shown in Figure 2.

7 Conclusions

Task distribution among multiple robots is a complex problem, which requires careful planning. The method presented in this work is capable of planning in an optimal manner, the sequence of movements of a multiple robot system in order to perform vision tasks. To accomplish this, the problem is stated as two different combinatorial optimization problems: activity assignment and tour planning. An evolutionary computation paradigm is adopted. The optimization method uses *random keys* representation along with *restriction-based heuristics* and auxiliary data structures. Experimentation results show the method's effectiveness and flexibility to changes in the environment. This work is incorporated into the photogrammetric network design system EPOCA, which can be considered as a new step towards the automation of high precision vision tasks. Additional work can be incorporated into the fitness function considering the robot motion-planning problem. Currently there are no obstacles incorporated within the simulation. In this way, collision between different robots, for example, are not considered. Also, aspects of the evolutionary process, as the size of the string representation, as well as incorporation of coevolution can be studied in future works.

Acknowledgements

This research was founded by contract 35267-A from CONACyT. Figures 1-3 were generated with software written at the Geometry Center.

References

1. S. Abrams, P.K. Allen, and K.A. Tarabanis, Dynamic sensor planning. In *IEEE International Conference on Robotics and Automation*, Atlanta, GA, 1993.
2. J. Bean, Genetic Algorithms and Random Keys for Sequencing and Optimization. In *ORSA Journal on Computing*, Vol. 6, No. 2, pp. 154-160. 1994.
3. E. Marchand and F. Chaumette, Active Vision for Complete Scene Reconstruction and Exploration, In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 1. January 1999.
4. G. Olague, Planification du placement de cameras pour des measures 3D de precision, PhD Thesis, Institut National Polytechnique de Grenoble, 1998.
5. G. Olague, Design and Simulation of Photogrammetric Networks using Genetic Algorithms, In *American Society for Photogrammetry and Remote Sensing*, 2000 Annual Conference Proceedings, 12 pages, Washington, DC.
6. G. Olague, Autonomous Photogrammetric Network Design using Genetic Algorithms, In E.J.W. Boers et al. (Eds.), Applications of Evolutionary Computing, Lecture Notes in Computer Science, Vol. 2037,pp. 353-363, Springer-Verlag, Lake Como, Italy, April 18 2001.
7. G. Olague and R. Mohr, Optimal Camera Placement for Accurate Reconstruction, *Pattern Recognition*, to appear. 2001.
8. Sakane S, Niepold R, Sato T, Shirai T (1992) Illumination setup planning for a hand-eye system based on an environmental model. In Advanced *Robotics*, Vol.6 No.4 pp.461--482.
9. Triggs B, Laugier C (1995) Automatic Task Planning for Robot Vision. In *International. Symposium of Robotics Research*, Munich.