Optimal Visual Sensor Placement using Evolutionary Algorithm

S.indu*, Chaitanya†, Manoj †, and Prof. Santanu Chaudhury† and Prof. Asok Bhattacharyya*
*Delhi College of Engineering
Bawana Road, Delhi-110042
Email: s.indu@rediffmail.com
†, Indian Institute of Technology,
Hauz Khas, New Delhi
Email: schaudhury@gmail.com

***Abstract*—Visual sensor arrays form the backbone of any multimedia and surveillance applications. This paper addresses the practical problem of optimally placing the multiple visual sensors satisfying the task constraints which may be static or dynamically varying according to requirements. We map this problem as an optimization problem using genetic algorithm by defining a coverage matrix as a function of set of sensor parameters and the space model parameters like priority, obstacles and feasible points, and solving the same to get the optimal set. The proposed method converges faster with lesser computational complexity compared to the linear programming approach and hence suited for surveillance of large spaces**

:

I. Introduction

Computer vision in video sensor networks has become a popular research topic in recent years. Decreasing cost of associated hardware and increasing practical need for such systems are among the reasons attracting more and more researchers to focus in this area. Different visual tasks have different requirements. Visual sensor network design supports applications such as intelligent rooms, video surveillance, automatic tracking etc. These applications require an efficient visual sensor layout which provides a minimum level of image quality or image resolution. An important issue of designing efficient visual sensor array is the placement of visual sensors (cameras) which optimizes the coverage of the specified area depending on the requirement of the user.This paper proposes a method to optimize position and poses of the visual sensor array to achieve maximum coverage of the specified priority area using genetic algorithm.

The main issue of offline camera placement is non availability of standard test beds for comparing the performances of different camera network setups. One of the main driving forces for this work is to study and improve the effect of the off-line camera placement on the machine vision systems on-line performance. Although significant amount of research exists in designing and calibrating video sensor arrays, automated visual sensor placement in general has not been addressed. There is some work in the area of grid coverage problems with sensors sensing events that occur within the circular range of the sensors [1]. In this paper we are using the mathematical model of a camera instead of circular range sensors. This paper maps the hard optimization problem of sensor placement into an application of genetic algorithm (GA). GAs are probabilistic search and optimization search techniques, which operate on a population of chromosomes, each representing a potential solution of the given problem, with aim to breed some high quality solution [2]. They are operationally simple and represent a good choice for solving problems with large search space, where only little is known about its characteristics and even for black box problems. As such they have been applied to many problems from the field of parameter optimization, planning and scheduling, design etc.

II. Related Work

Significant amount of research has been made in solving optimal guard location problems for a polygonal area, e.g., The Art Gallery Problem (AGP) and its variants, where minimum numbers of Guards are determined so that all points of the polygon can be observed for their static positions. The exact solution of the same is found to be NP-Hard, even though efficient algorithms exist giving a lower bound for AGPs with simple polygons [3], [4], [5], [6]. Current solutions to the AGP and its variants employ unrealistic assumptions about the cameras capabilities like unlimited field of view, infinite depth of field, infinite servo precision and speed that make these algorithms unsuitable for most real world computer vision applications. These Work has been done from the planning perspective for control of sensors, their coverage, fusion and deployment but however they have failed to explicitly incorporate a pragmatic camera model. While considering task constraints a practical model has been developed to address dynamic tasks as well, it however employs a binary 0-1 optimization model that has limited implications and becomes too tedious for any practical realizations. [2]was a classic treatise and has been very handy in completely redefining the optimization problem.

A. Mittal et.al [7] and Ugur [8] did a probabilistic approach to maximize the coverage of a specified area using multi cameras, by optimizing the poses of the cameras with fixed locations. E. Horster et al [9] have considered a more realistic model of camera and they adopted the binary integer

programming method for optimization. The BIP method is computationally complex and may not give accurate solution for complex spaces. Both the above said approaches have not considered 3D coverage space.

G. Olgue [10] and [11] used Evolutionary computing method for Photogrammatic Netework design, where the positions of the cameras were fixed and the pan and tilt angles were varied to get the optimum poses. The method also assumes a predefined shape for the objects to be photographed. In the proposed method we optimise both position and poses simultaneously using Genetic Algorithm. We divided the area to be photographed as uniform grids and classified each one as Priority point, Non-priority point or Obstacle.

III. Problem Formulation

A. Problem Statement

To determine the optimal locations of the pan-tilt cameras and their poses to achieve maximum coverage for a given a floor plan, with some obstacles and priority as well as non-priority areas. We consider both the offline and the online placement schemes using pan-tilt cameras in this paper. These cameras continuously rotate about their pan and tilt axes, thus the effective area covered by a single camera increases drastically, which decreases the number of cameras required to cover the given floor plan.

We divide the given floor plan into priority and non-priority areas. Since the cameras are in continuous motion about their pan and tilt axis, the time for which a certain area is in focus is limited. But we want a priority area should be on focus for at least a fixed proportion of time. For achieving this, we define a priority area as covered only when it lies in the intersection of the extended field of view of certain number of cameras. This can be understood from the following fig. 1. Here two

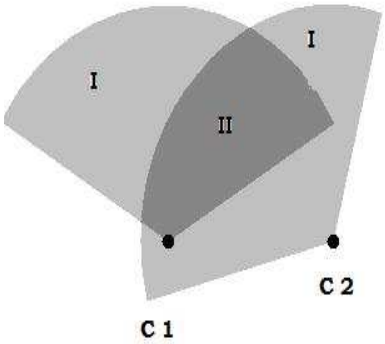


Fig. 1. 2D overlapping field of View

cameras are placed at c1 and c2. The shaded region represents the extended field of view. Area lying inside region II will be periodically covered by both the cameras and hence it will be visible for more proportion of time than region I. Thus priority areas should lie inside region II.

The cameras cant be practically placed anywhere in the floor plan. There are certain areas which we name as feasible areas where camera placement is practically possible. The proposed method places cameras to achieve maximum coverage in the user defined feasible area. The method also considers obstacles in the floor plan while determining the location of cameras.

B. Modeling of the Floor Plan

We model the floor plan by means of grid points. The grid points are separated by a fixed distance. The grid points lying inside priority areas are termed as priority grid points while others are non priority grid points. Areas where camera placement is feasible are also modeled through grid points and these are termed as feasible points. Thus the problem now becomes a grid coverage problem.

In ideal cases, the poses of pan tilt camera are infinite corresponding to the infinite values of pan and tilt angles which the camera can assume. As we are not able to solve our problem for continuous case we approximate the continuous case by sampling the poses. Cameras can only adopt those discrete and finite numbered poses.

Coverage Metric: We define coverage to be:

$$\mathcal{C} = \alpha \sum_{priority} t_i + \sum_{non-priority} m_j \tag{1}$$

Where

$$t_i = \begin{cases} 1 & if\ priority\ point\ i\ is\ covered\ by \\ & atleast\ two\ cameras \\ 0 & else \end{cases}$$

And

$$m_j = \begin{cases} 1 & if\ non-priority\ point\ j\ is \\ & covered\ by\ atleast\ one\ cameras \\ 0 & else \end{cases}$$

Here t represents priority points while m represents non priority points.α is the priority parameter which decides the weight to be given to the coverage of priority points over non priority points. α =1 corresponds to equal weight. Better solutions will have a higher value of C.

Having formulated a performance metric of solution, we can consider it as an optimization problem where C has to be optimized.

C. Proposed Approach for Optimal Camera Placement

We have considered floor plan to be a cube which can be represented by NxNxN grid points. We then define the priority points, feasible points and obstacles in terms of matrices. Matrix P denotes the priority points and is defined as:

$$\mathcal{P} = [P_{ijk}]_{NxNxN} \tag{2}$$

Where

$$P_{ijk} = \begin{cases} 1 & if\ i\ ,j\ ,k\ point\ is\ a\ priority\ point \\ 0 & if\ i\ ,j\ ,k\ point\ is\ anonpriority\ point \end{cases}$$

Similarly, we define feasible matrix as feasible locations of cameras

$$\mathcal{F} = [f_{ijk}]_{NxNxN} \tag{3}$$

Where

$$f_{ijk} = \begin{cases} 1 & if\ i\ ,j\ ,k\ point\ is\ a\ feasible\ point \\ 0 & if\ i\ ,j\ ,k\ point\ isnot\ afeasible\ point \end{cases}$$

And obstacle matrix as

$$\mathcal{O} = [f_{ijk}]_{NxNxN} \quad (4)$$

Where

$$o_{ijk} = \begin{cases} 1 \quad if\ i\ ,j\ ,k\ lies\ in\ obstacle\ region \\ 0 \quad if\ i\ ,j\ ,k\ does\ not\ \ lies\ in\ obstacle\ region \end{cases}$$

After the generation of these matrices, the visibility matrix is generated which is 3 Dimensional globally and denotes all the grid points that are covered by a given camera placed at a feasible point with a particular pan and tilt angle. It takes care of obstacles as if a point is occluded from a camera by an obstacle then the coverage matrix will note that and report the point to be uncovered by the particular camera.

While defining the criteria for the coverage of a point by a camera, we considered certain sets of inequality that define the volume covered by the camera having the particular pose. But we have considered the camera to be continuously rotating about its pan and tilt axes, and thus to incorporate this in our model, we increase the 2D projected cones angle by max pan limit angle for the YZ projection and the max tilt limit angle for the XZ projected cone. Any point lying in this extended region is covered by the camera periodically, and thus we consider the point to be covered by a camera.

The visibility matrix when simply generated comes out to be 8 Dimensional which is very inconvenient to work with. Thus we map every grid point to a particular number by using the method by Ugur Murat Erdem et. al [8]

$$position(i,j,k) = (j-1) * N * N + (i-1) * N + k \quad (5)$$

And similarly every pose by the mapping:

$$pose(\alpha, \beta) = M * (\alpha - 1) + \beta \quad (6)$$

Where M is the no. of discrete pan or tilt angles the camera can assume.

The visibility matrix is now reduced to a 3 Dimensional matrix which can be expressed as

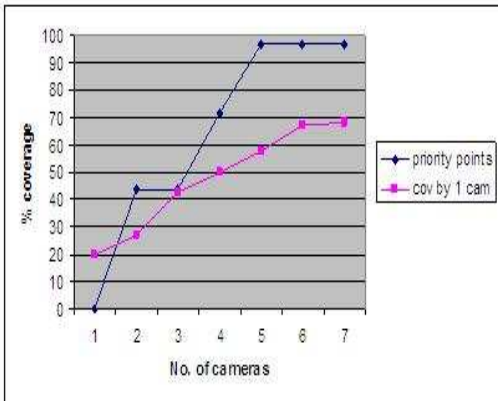
$$A = [a_{ijk}]_{N^3 xM^2 xN^3} \quad (7)$$

Where

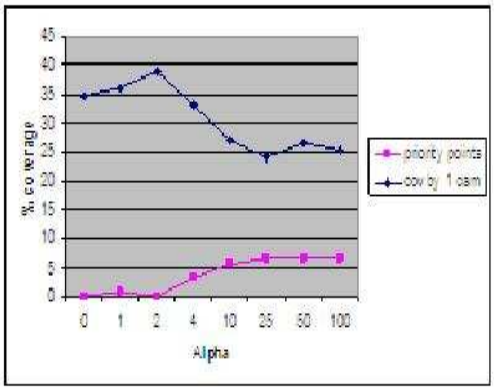
$$a_{ijk} = \begin{cases} 1 \quad if\ camera\ at\ point\ i\ with\ pose\ j\ covers\ point\ k \\ 0 \quad else \end{cases}$$

This visibility matrix along with the priority matrix is then used to calculate the coverage score of any set of cameras placed at different locations. For simulation purposes, we consider the floor plan to be a simple cube with each side being covered by 10 grid points. Also all the sides of the cube except the floor are being considered as a feasible camera location. The priority area is considered to be a smaller 4x4x4 cube with its center coinciding with the center of the floor plan. For the obstacle, we consider it to be a pillar extending from grid points 3 to 5 in x, y directions from floor.

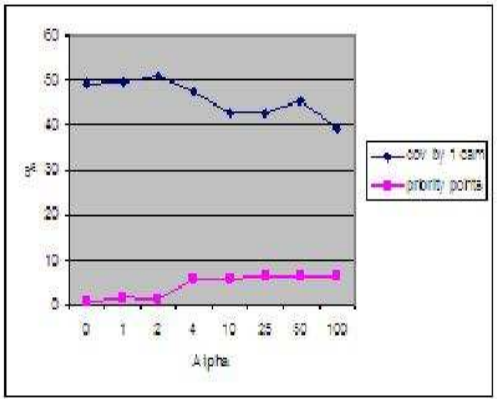
We propose the use of Genetic Algorithm to solve this hard optimization problem. The visibility matrix and the priority matrix help the Genetic Algorithm to evaluate the fitness function of various generations, which here is the coverage score.



(a)



(b)



(c)

Fig. 2. Shows percentage coverage by varying (a) number of cameras (b) α and using only 2 cameras (c) α and using only 3 cameras

D. Genetic Algorithm Mapping

The problem addressed in this paper is to find an optimal configuration of an array of visual sensors such that they satisfy the space constraints and to make total coverage maximum. The first and the foremost step of a design of a genetic algorithm is to select the variables of the problem

to be solved. This is a crucial point since on this selection other features of the algorithm depend. The variables should represent size of the search space, efficiency of the genetic operators etc. The most natural way of representing solutions of this problem would be a sequence of genes, each coding the actual position and the pose of individual camera.

Optimization criteria: max

A simple way of encoding would be through a binary bit string:

$$Gene\ of\ a\ cameraC(i) = (X(i), Y(i), Z(i)\alpha(i), \beta(i)) \quad (8)$$

where
$X(i)$= $\{\ a_1, a_2, ...a_k\}_{10}$
$Y(i)$= $\{\ b_1, b_2, ...b_k\}_{10}$
$Z(i)$= $\{\ c_1, c_2, ...c_k\}_{10}$
$\alpha(i)$= $\{\ h_1, h_2, ...h_s\}_{10}$
$\beta(i)$= $\{\ j_1, j_2, ...j_s\}_{10}$
K= $log_2(N)$.
where coordinate feasible space is of dimension N^3
i.e. 0$\leq$ x[i],y[i],z[i] $\leq$ N-1
and s = $log_2(N_0)$;
where pan-tilt space is of cardinality
i.e. $N_0 \leq \alpha$[i],β[i] $\leq N_0$-1
$\{\ ..,..,..\}_{10}$ is decimal representation of a binary bit string with left most bit as MSB.

For the one-to-one mapping

$$position(i, j, k) = (j-1) * N * N + (i-1) * N + k \quad (9)$$

$$pose(\alpha, \beta) = M * (\alpha - 1) + \beta \quad (10)$$

The gene of each camera C[i], is simply a concatenation of two bit strings.

Alternatively speaking, the gene of camera is an abstraction of its location and orientation of its pose in the space. Being a collection of genes, a chromosome would therefore be a representation of an array of cameras belonging to the solution space. So our problem becomes redefined to look into the solution space and choose the fittest among them. The fitness function very obviously is the coverage metric for each set of cameras.

E. Algorithm

1) An initial random population of N belonging to the search space (within the feasible region only) is chosen and encoded by the above procedure.
2) Next we evaluate the fitness value for each of the population using the matrices of coverage of priority and non priority points generated and a comparison is made regarding the optimality of the solution.
3) Then, we select a population of good networks by tournament selection method, two best individuals are simply passed on and we proceed for reproduction.
4) From this population we recombine the species using the following operations:

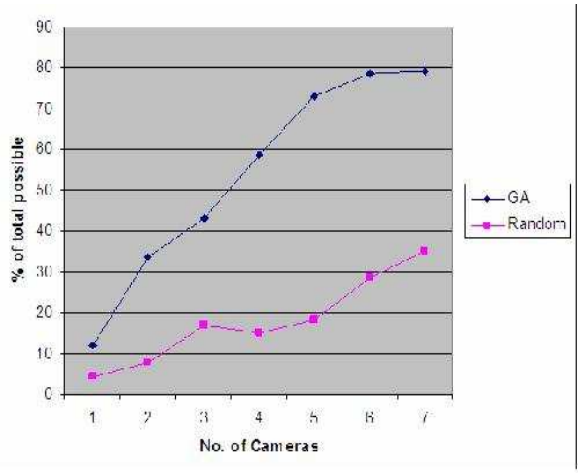


Fig. 3. GA vs random placement

   a. Crossover with a probability of 0.8 using scattered crossover function.
   b. Mutation with a probability of 0.001 is essential to maintain diversity.
5) These operations yield a new population which replaces the existing one.
6) Steps 2, 3, 4 are repeated until the optimization criterion stabilizes.

All the above implementation has been achieved through the GA package (and toolbox) provided with MATLAB Version 7.0. We used the camera model developed by E.Horster et.al [9]

IV. Simulation Results

All the coding and matrix representations have been implemented in Matlab. For the purpose of drawing 2-D spaces we have used the help of JAVA- 2D classes to visualize our task. In the case of 2D for simplicity we have considered the camera field of view to be an arc of variable subtended angle and feasible space containing the whole floor plan. while in case of 3D a simple cube has been considered.

The graph shown in fig. 2 (a) shows that we require only 4 cameras to cover the specified area and fig. 2 (b) and fig. 2 (c)shows that the max value we can select for weight-age(α)is 10.The graph shown in fig. 3 shows the coverage variation by random placement of cameras and Placement using GA The positions and poses of camera when we use 3D model and 2D model are shown in fig. 4 and fig. 5 respectively

V. Conclusions

We have presented a novel method of optimizing visual sensor array placement by simultaneously adjusting their poses and positions to get maximum coverage using Genetic Algorithm. Our method allows for a trade off between accuracy of results obtained and time taken as when not very accurate results are required, the GA can be stopped and the fittest entity can be considered as the solution. The method also is

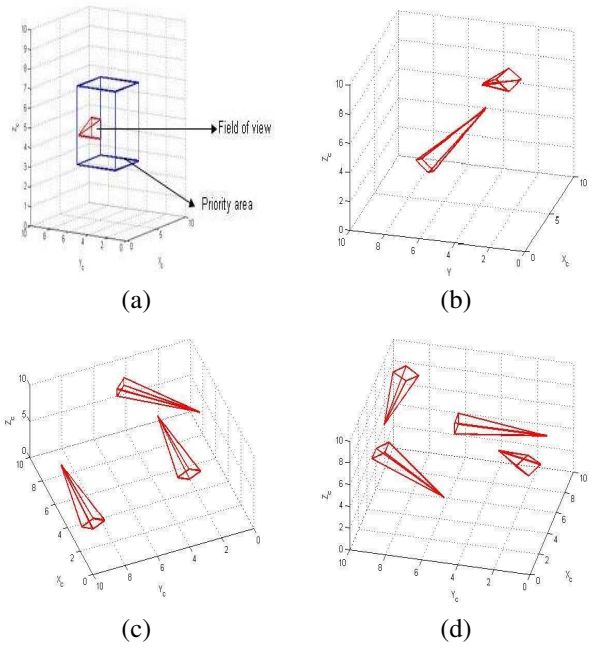


Fig. 4. Shows the position and pose of the camera to cover a volume (a) using 1 camera (b) using 2 cameras (c) using 3 cameras (d) using 4 cameras

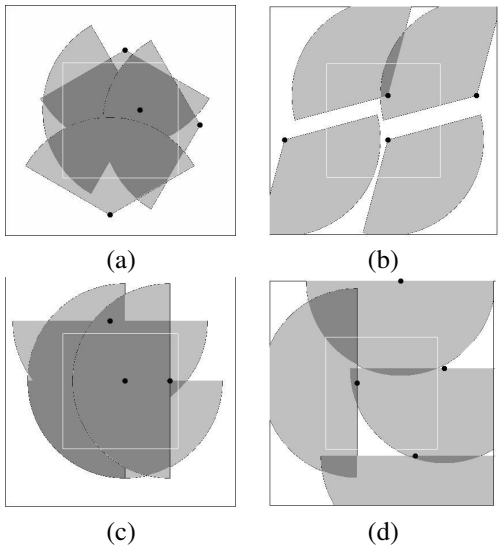


Fig. 5. Shows the position and pose of the camera to cover an area, for (a) and (c) Inner square as the priority area and for (b) and (d) Equal priority for inner and outer square

computationally lighter for higher values for grid points than methods involving techniques like Linear Programming. Also, having considered details like priority points, feasible camera placement locations, and obstacles, our solution is practically implementable.

We have considered periodic motion of pan tilt camera, resulting in extending the field of view of the camera and this allows the method to give very economical solution in situations where continuous coverage will result in unnecessary surveillance redundancy.

VI. Future Work

We have not considered the motion model of the camera in this paper. Consideration of motion trajectory of the camera is important in application involving trajectory tracking of objects. The method when run for high number of grid points becomes memory inefficient, this area as to be looked upon more closely.

References

[1] K.chakraborty, S.S.Iyengar, H.: Grid Coverage for Surveillance and Target Location in Distributed sensor Networks. Computer Vision and Image Understanding **51(12)** (2002) 1448 – 1453

[2] Goldberg, D.V.: Genetic Algorithm in Search Optimization and Machine Learning. Addison Wesley Longman Publishing Company Ltd (1997)

[3] Rourke, J.O.: Art Gallery Theorems and Algorithms. Oxford University Press (1987)

[4] I. Suzuki, Y. Tazoe, M.Y.: Searching a polygonal region from the boundary. Int. J. Comp. Geom. **5** (2001) 529 – 553

[5] P. Bose, L.J. Guibas, A.L.M.H.O.D.L.S.J.U.: The Flood Light Problem. Int. J. Comp. Geom. **1** (1997) 153 – 163

[6] V. Estivill-Castro, J. O'Rourke, J.U.D.X.: Illumination of polygons with vertex lights. Information Process Letter 56 **1** (1995) 9 – 13

[7] A. Mittal, L.S.D.: Calibrating and optimizing Poses of Visual Sensors in Distributed Platforms . In: European Conference on Computer Vision (ECCV). (2004)

[8] Erdem, U.M.: Stan Sciaroff, Automated Camera layout to satisfy task specific and floor plan specific coverage requirements . Computer Vision and Image Understanding **103** (2006) 156 – 169

[9] Horster, E., Lienhart, R.: Illumination of polygons with vertex lights. ACM Multimedia systems journal, Special issue on Multimedia Surveillance Systems (2006)

[10] Mohr, R.: Optimal Camera Placement for Accurate Reconstruction. Technical Report 3338 (January 1998)

[11] Dunn, E.: Development of a practical Photogrammatic Network Design using Evolutionary Computing. Photogrammatic Record **17** (October 2006)