PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04N

A2

(11) International Publication Number: **WO 00/56056**

(43) International Publication Date: 21 September 2000 (21.09.00)

(21) International Application Number: PCT/US00/40011

(22) International Filing Date: 17 March 2000 (17.03.00)

(30) Priority Data:
60/124,931 18 March 1999 (18.03.99) US

(71) Applicant *(for all designated States except US)*: SHOWBITES, INC. [US/US]; 1209 Orange Street, Wilmington, DE (US).

(72) Inventors; and

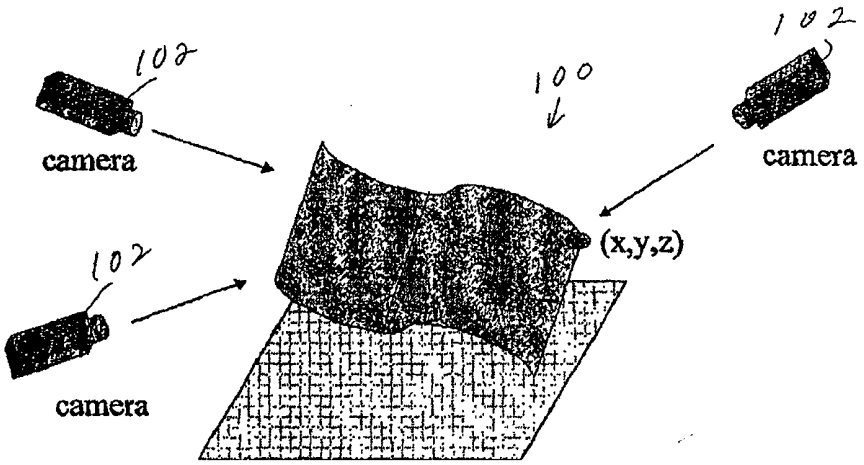
(75) Inventors/Applicants *(for US only)*: LEVIN, Moshe [IL/IL]; 12 Rekanati Street, 69494 Tel Aviv (IL). BEN MORDECHAI, Ido [IL/IL]; 14b Hadaliot Street, Ramat Poleg, 42650 Natania (IL).

(74) Agents: COHEN, Herbert et al.; Blank Rome Comisky & McCauley LLP, The Farragut Building, Suite 1000, 900 17th Street, NW, Washington, DC 20006 (US).

(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: METHOD FOR OPTIMIZATION OF VIDEO COVERAGE



(57) Abstract

The positions and angular orientations of cameras used to cover a predetermined volume such as a hall or sports field are optimized by a combination of a genetic algorithm and a simulated annealing algorithm. In the genetic algorithm, first, random initial solutions are generated. A local search is performed around each of the random initial solutions to find a locally optimized solution. A random mutation is applied to each of the locally optimized solutions to obtain a mutated solution. The mutated solutions are recombined to obtain recombined solutions, which are sorted by coverage level. The recombined solutions having the highest coverage levels are selected for the simulated annealing algorithm. The simulated annealing algorithm begins by randomly generating a new solution which is separated from the recombined solution by less than a predetermined search radius. A coverage level is calculated for the new solution. The simulated annealing algorithm is reiterated until a global, rather than merely local, optimization is achieved. In the algorithms, each solution at each state is represented by a matrix whose orders equal the number of cameras to be placed and the five degrees of freedom of each camera (three coordinates of position and two coordinates of angular orientation). The mutation can also be used to optimize the number of cameras.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

METHOD FOR OPTIMIZATION OF VIDEO COVERAGE

REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Application No. 60/124,931, filed March 18, 1999, whose disclosure is hereby incorporated by reference in its entirety into the present disclosure.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to video coverage of a given space using fixed cameras in which each camera has a predetermined three-dimensional orientation (azimuth and elevation) and more particularly to a method of optimizing the placement and angular orientation of the cameras.

Description of the Related Arts

Real and still remote video systems cover a given space by a multiplicity of video cameras. The video cameras may be fixed or mobile. Firstly, each camera has a location and angular orientation that may be defined by the vector (x, y, z, θ, ϕ) where x, y, z are the three-dimensional coordinates and θ and ϕ are the azimuth and elevation of the camera line of sight are relative to the x, y, z coordinate system. Secondly, we will define $\alpha_{\theta,i}$ and $\alpha_{\phi,i}$ as the maximum field of view for each camera and c as a cost attribute. We will define the space to be covered as a three-dimensional surface of points $x_k y_k$ defined by a height coordinate z for each $x_k y_k$. A set of N cameras will be defined by M, an $N \times 8$ matrix. Mobility is defined as a change in any one of the above mentioned vector (or matrix) values. The level of mobility is determined by the mechanical mounting of the camera.

A coverage priority level is defined such that for each point (x_j,y_j,z_j) a priority of coverage value, P_j, is assigned.

Further we define a coverage level criterion *CL* that for each given matrix M represents the quality of coverage.

The prior art provides no efficient way to optimize *CL*. Trial and error should not be relied upon.

SUMMARY OF THE INVENTION

It is therefore an object of this invention to determine the best location and angular orientation for a given set of cameras, best covering a predetermined three-dimensional area. The invention is based on building a computerized three-dimensional model (like a DTM map) of the area to be covered, setting coverage priorities to each point on the three-dimensional surface and using an efficient combination of mathematical optimization techniques to set the best location of cameras required to obtain a predefined level of quality.

It is a second object of this invention is to determine the best subset of cameras, locations and orientation, chosen from a predefined set of cameras. That is, the number of cameras is optimized along with their positions and angular orientations. The selected set will have minimal total cost while maintaining a predefined criteria of coverage.

The present invention provides a method of selecting a plurality of fixed camera locations and angular orientations and which in one embodiment is capable of reducing the total required number of cameras to cover a given three dimensional space. Each single camera has a mechanical installation point and fixed orientation (elevation and azimuth angles relative to a chosen coordinates system). The cameras are capable of optical zoom in and zoom out and electronic selection of the view within the mechanical limitation of the camera mount. The method determines the number of cameras required to cover a

predetermined space (hall, sports field or any other defined location) within required performance criteria. Further, the method provides a mechanical mounting location for each of the selected cameras, and the installation orientation.

The cameras can be of the sort that have electronic navigation capabilities only. This invention can be used with remote access real and still video applications.

The optimization problem for a given space is to find the number N of cameras out of the set of $K \geq N$ cameras, and the values $(x,y,z,\theta,\phi,\alpha_{\theta,i},\alpha_{\phi,i},c_i)$ $0<i<N$ which will meet a predetermined level of CL. In addition, the optimization technique will optimize the coverage for a given level of coverage.

The optimization technique utilized is a combination of genetic algorithms and simulated annealing.

Genetic algorithms are local search methods where the neighborhood generation mechanism is inspired by real life process of genetics and evolution. In particular, the current sub-optimal solution is modified by "splicing" and "mutation" to obtain the next generation solutions.

Simulated annealing is a simulated technique, which allows to provide an efficient search with a mechanism allowing not to be stuck in one local minimum.

The combination allows exploitation of both optimization techniques to produce a robust fast converging solution.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects, features and advantages of the present invention will become apparent from the following description, taken in conjuncture with the accompanying drawings in which:

Fig. 1 is a view for describing the three-dimensional model of the area to be covered by the plurality of cameras.

Fig. 2 is a view of the three-dimensional model describing the ceiling of the above model. The ceiling here means the maximum allowable height of the camera.

Fig. 3 is a view of all the parameters of the camera.

Fig. 4 depicts a high-level flowchart of the suggested method.

Fig. 5 depicts the camera footprint calculation.

Fig. 6 describes the recombination process.

DETAILED DESCRIPTION OF THE INVENTION

Invention Overview

The present invention will hereafter be described with reference to the accompanying drawings.

Fig. 1 is a view for describing the three-dimensional model 100 of the area to be covered by the plurality of cameras 102. For each pair of coordinates x, y we define a height z. The vector (x,y,z) defines a point in space which is the surface to be covered. The surface here can describe the ground level or the outside of the building which the system has to cover.

Fig. 2 is a view of the three-dimensional model 200 describing the ceiling 202 of the above model 100. The ceiling here means the maximum allowable height of the camera. This value can be set from the actual height of the area roof, the maximum height determined by safety reasons, the maximum height of a camera carrying construction. The ceiling is defined by the vector (x,y,z_m), which sets the boundary conditions for the optimization algorithm procedures.

Fig. 3 is a view of all the parameters of the camera 102:

x	The x location of the camera.
y	The y location of the camera.
z	The z location (height) of the antenna.
θ	The azimuth of the camera, relative to the x axis.
ϕ	The elevation of the camera, relative to the x,y plane.

Fig. 4 is a high level description of the method according to either of the preferred embodiments to be set forth below. First we input a three-dimensional view of the area to be covered which is the set of points (x,y,z) (step 402). Then we input the ceiling of the area which sets the boundary condition for the location of the cameras (x,y,z_m) (also step 402). Further, we input the coverage priority for each point on the surface P(x,y,z) (step 404).

The next step is to define the optimization parameters (details are discussed later) (step 406). These parameters include (but are not limited to): number of iteration for the genetic algorithms, algorithm parameters, number of output solutions, number of iteration of the simulated annealing simulation parameter.

The genetic algorithm is activated to produce several solutions (step 408). For each solution simulated annealing technique will be used to improve the quality of the solution (step 410).

Fig. 5 describes the method by which the coverage of a single camera is calculated. According to the view field, the presence or absence of each line of sight 502 is analyzed from each point in the area.

Three-dimensional model of the area to be covered

The areas to be covered are defined by a three-dimensional model. The model is a set of a finite number of vectors (or points in space). The model is typically (but not necessarily) defined by a polygon that bounds the x,y plane and a resolution which sets the horizontal

distance between any two adjacent points. The same structure applies to the ceiling of the model which can be viewed as a second layer in a three-dimensional map and to the priority of coverage criteria which is attached to each three-dimensional point.

Higher quality results can be obtained if a second layer of "roof level" polygons is available in a matrix format. Such a layer can improve significantly the accuracy of the model, and hence the quality of the coverage.

An intermediate set is defined as L(x,y,z). For each point in the model, and a given set of cameras position and space orientation, L(x,y,z) is defined as the number of cameras that have lines of sight to (x,y,z). Under this definition L(x,y,z)=0 means that the specific point is not covered by any of the cameras.

Coverage quality calculation

The coverage quality will be calculated by the following formula:

$$CL = \frac{\sum_{x,y,z} L(x,y,z)^{k1} \cdot P(x,y,z)^{k2}}{\sum_{x,y,z} P(x,y,z)^{k2}}$$

where k_1 and k_2 are calibration factors used to calibrate the relevant importance of the various points. A large k_1 optimizes the coverage with emphasis on coverage diversity, i.e., several cameras for each point on the model. A small k_1 optimizes the system to have at least one camera over each point. Similarly, a large k_2 optimizes the coverage with strong advantage to the high priority points, while a small k_2 lessens the impact of priority for a given coverage.

Two preferred embodiments will now be set forth. In the first, the number of cameras is given; in the second, the optimization determines a cost-effective number of cameras.

1) **Optimization method: a given set of cameras**

Setting global optimization parameters

The optimization parameters to be set are:

Iter_genetic	number of genetic algorithm iterations
Num_of_iter_solutions	number of solutions (M matrixes) that will be generated by the genetic algorithm.
Iter_annealing	number of simulated annealing algorithm iterations (for each of the Num_of_iter_solutions.

For iter_genetic=0, the optimization procedure will not include the genetic algorithm phase and the required number of intermediate number of solutions determined by the parameter num_of_iter_solutions will be generated automatically.

If the number of the simulated annealing iterations as defined by the parameter Iter_annealing is set to zero, the parameter Num_of_inter_solutions will automatically be set to 1. The selected M matrix will be the result of the genetic algorithm.

Genetic algorithm to optimize three-dimensional coverage for a given set of cameras

Algorithm local parameters:

Size_of_population - number of feasible solutions to be used in the simulation

Differential_step - (dx_i, dy_i, dz_i, $d\theta_i$, $d\phi_i$) the differential step for each variable

Local_search_iteration - number of local search iterations

mutation_probability - the probability that a specific value will be changed in the mutation phase.

Urv - intermediate parameter, random variable.

power_of_bias - the urv defined earlier is raised to the power of power_of_bias and provide adjustment to the level of randomness in the selection process. For power_of_bias=0 no randomness is introduced to the process.

The algorithm's flow

Step 0 - init:

Define randomly a set of size_of_population M feasible matrixes.

Repeat steps 1-4 iter_genetic interations

Step 1 - local search

For each matrix M out of the size_of_population number of matrixes, repeat.

Local_search_iteration number of iterations:

1.1 Calculate the coverage level indicator CL

1.2 Calculate the coverage level indicator CL for (N*5) differential moves, each defined by the size of the relevant value in the differential_step parameter.

1.3 Change the value of the matrix which generated the higher level of CL, the change of the value will be in the size of: new_value = old_value - differential_step*(CL_{new}-CL_{old}), where the differential step is selected to the appropriate variable under consideration.

1.4 If (CL_{new}-CL_{old})=0, stop.

At the end of this step we have size_of_population.

Step 2 - Mutation

For each value of each matrix (total of size_of_population*N*5 values) determine with probability level of mutation_probability whether this specific value will be changed.

For each of the to be changed values, calculate a new matrix with the specific value according to the following equation.

New_value = Old_value + random number.

Step 3 - Recombination

Randomly select a number_of_combination subset of matrix pairs (M_1,M_2). From each pair of matrixes create a new matrix according to any of several combination mechanisms. One of them is described in the following recombination rules, describing the "breeding" process:

3.1 Calculate N integer random variables B_i, $1<i<5$.

3.2 The new matrix will be constructed out of the pair where for each vector (camera location) the first B_i elements will be taken from the matrix M_1 and the following 5-B_i elements will be taken from the matrix M_2.

Fig. 6 depicts the recombination process for a matrix of 5 cameras.

Step 4 - Survivor selection

Steps 2 and 3 created a set of up to $2S+S^2$ possible matrixes which stands for the cameras' positioning and orientation, where S stands for the size_of_population variable. Out of this population we select a next population of size_of_population solutions or cameras positioning according to the following procedure.

4.1 For each matrix calculate CL.

4.2 Multiply CL by a random bias factor: $CL_B = CL*(urv^{power_of_bias})$

where urv is a random variable distributed uniformly over the range (0,1] and power_of_bias controls the effect of the random variable. If power_of_bias = 0, then CL_B=CL and there is no effect to the random factor.

4.3 Sort the population according to CL_B.

4.4 Select the size_of_population highest ranking variables as the next generation population.

Step 5 - Selection of intermediate solutions

If the number of iteration for the simulated annealing is set to zero (iter_annealing=0), the highest rank solution (matrix) is the selected solution for the given number of cameras.

If the number of iterations is higher than zero, then from the population of size_of_population matrixes, select num_of_iter_solutions highest ranking matrixes as input to the next phase.

Simulated annealing procedure to further improve three-dimensional coverage

Algorithm local parameters:

T = temperature

Search_radius = search radius around the reference solution.

The algorithm's flow

For each of the num_of_iter_solutions matrixes repeat the following steps inter_annealing iterations:

Step 1 - Generate randomly a matrix in a distance (per variable) less than search_radius from the reference matrix.

Step 2 - Calculate the resulting coverage level CL_{new}.

Step 3 - if CL_{new} is lower then CL then the new matrix will be the new reference. Otherwise, the new matrix will be selected as the reference matrix with a probability of $e^{-(CLnew-CL)/T}$. This selection mechanism allows actual searching of global minimum and prevents locking in local minimum.

2) Optimization method: selecting a subset of cameras to meet a required coverage level

In this case, the genetic algorithm is modified to handle selection of a subset of cameras out of a given set of cameras. Various cost functions can be used to sort the tested matrixes. In the following section, we describe the modified genetic algorithm. We

essentially describe the difference between the algorithm described earlier and the required algorithm.

The coverage-cost criteria

Discussing selection of a subset of cameras with several cost levels we have to modify the coverage level to include costs:

$$CL_C = \left(\frac{CL}{CL_0}\right)^{r_1}\left(\frac{TC_0}{TC}\right)^{r_2}$$

where:

- CL_C is the generalized quality of coverage.
- CL is the previously defined quality of coverage.
- CL_0 is a threshold level of CL.
- TC is the total cost of the cameras.
- TC_0 is a threshold level of the cost.
- r_1 depicts the relative influence of the quality of coverage as measured earlier. For $r_1=0$ the actual cost of the cameras will be the only factor. For $r_1>>1$, CL values higher than CL_0 have significant advantage.
- r_2 depicts the relative influence of the cameras cost. For $r_2=0$, the quality of coverage will be the only factor. For $r_2>>1$ TC values lower than TC_0 have significant advantage.

The algorithm's flow

Step 0 - init:

Define the set C of available cameras, each with 6 parameters (adding camera cost to the previous definition)

Define N as an initial guess on the required number of cameras to meet the coverage generalized quality criterion CL_C.

Define randomly a set of size_of_population feasible matrixes each with N vectors (N cameras).

Set the two-dimensional optimization parameters r_1,r_2.

Repeat steps 1-4 inter_genetic iterations.

Step 1 - local search

This step remains unchanged since the local search is done on the per matrix calculation and the cost component is not relevant here.

For each matrix M out of the size_of_population number of matrixes, repeat Local_search_iteration number of iterations:

1.1 Calculate the coverage level indicator CL.

1.2 Calculate the coverage level indicator CL for (N*5) differential moves, each defined by the size of relevant value in the differential_step parameter.

1.3 Calculate the cost of the system: the sum of the sixth column update.

1.4 Change the value of the matrix which generated the higher level of CL, the change of the value will be in the size of: new_value = old_value - differential_step*(CL_{new}-CL_{old}), where the differential step is selected to the appropriate variable under consideration.

1.5 If (CL_{new}-CL_{old})=0, stop.

At the end of this step we have size_of_population.

Step 2 - Mutation

This step is the first step in the creation of matrixes with different sizes (variable numbers of cameras).

2.1 Location Change: For each value of each matrix (total of size_of_population*N*5 values) determine with probability level of mutation_probability whether this specific value

will be changed. For each of the values to be changed, calculate a new matrix with the specific value according to the following equation: New_value=Old_value + random number.

2.2 Add/Remove Cameras: Generate a random number out of a normal distribution with average μ and standard deviation σ, add this number for the current number of cameras in the matrix and round the results. If the addition results is lower than the current number of cameras, select randomly the required number of cameras and remove. If the addition result is higher than the current number of cameras, add the required number of cameras selecting randomly from the set of feasible cameras C.

Step 3 - Recombination

This step creates matrixes with different sizes (variable numbers of cameras).

Randomly select a number_of_combinations subset of matrix pairs (M_1,M_2). M_1 and M_2 can be of different sizes, N_1 and N_2. From each pair of matrixes create a new matrix according to any of several combination mechanisms which can change the size of the result matrix. One of them is described in the following recombination rules:

3.1 Generate randomly N_{new} the number of cameras in the result matrix M. Without loss of generality we assume: $N_1<N_{new}<N_2$.

3.2 Calculate N_{new} integer random variables B_i, $1<i<5$.

3.3 The new N_1 elements of the matrix will be constructed out of the pair where for each vector (camera location) the first B_i elements will be taken from the matrix M_1 and the following $5-B_i$ elements will be taken from the matrix M_2.

3.4 The remaining $N_{new}-N_1$ elements will be constructed by randomly "breeding" the $N_{new}-N_1$ elements of N_2 with random selection of vectors in M_1.

Step 4 - Survivor selection

The above described process results in a variable size population, i.e., coverage solutions with variable number of cameras. The survivor selection handles matrixes with variable numbers of cameras as well.

Steps 2 and 3 created a set of size of up to $2S+S^2$ possible matrixes which stand for cameras positioning and orientation, where S stands for the size_of_population variable. Out of this population, we select a next population of size_of_population solutions positioning according to the following procedure:

4.1 For each matrix calculate CL_C.

4.2 Multiply CL_C by a random bias factor: $CL_{BC} = CL_C*(urv^{power_of_bias})$ where urv is a random variable distributed uniformly over the range (0,1], and power_of_bias controls the effect of the random variable. If power_of_bias=0, then $CL_{BC}=CL_C$ and there is no effect to the random factor.

4.3 Sort the population according to CL_{BC}

4.4 Select the size_of_population highest ranking variables as the next generation population.

Simulated annealing procedure

The simulated annealing procedure is done separately on each of the genetic algorithm suggested matrixes. As a result, these procedures do not need to handle variable sizes of matrixes.

The above procedures can be carried out on any suitable computing device, such as a suitably programmed IBM-compatible microcomputer. An output (e.g., a display or a printout) can provide the optimized positions and angular orientations of the cameras, as well as the optimized number of cameras in the second embodiment. Those skilled in the art who

have reviewed the present disclosure will readily appreciate which hardware and software are required; therefore, such details will not be set forth here.

While two preferred embodiments of the present invention have been set forth above, those skilled in the art who have reviewed the present disclosure will readily appreciate that other embodiments can be realized within the scope of the present invention. For example, while specifics of the genetic and simulated annealing algorithms have been set forth, other appropriate algorithms can be used instead. Therefore, the present invention should be construed as limited only by the appended claims.

What is claimed is:

1. A method for determining locations and angular orientations of a set of cameras to cover a predetermined volume, the method comprising:

(a) determining a number of said cameras;

(b) determining at least one intermediate solution through a genetic algorithm; and

(c) determining a solution from the at least one intermediate solution through a simulated annealing algorithm.

2. The method of claim 1, wherein step (b) comprises:

(i) determining a plurality of random initial solutions;

(ii) performing a local search around each of the random initial solutions to find a locally optimized solution;

(iii) applying a random mutation to each of the locally optimized solutions to obtain a mutated solution;

(iv) recombining the mutated solutions to obtain recombined solutions;

(v) sorting the recombined solutions by coverage level; and

(vi) selecting a number of the recombined solutions having the highest coverage levels for the simulated annealing algorithm.

3. The method of claim 2, wherein step (c) comprises, for each of the recombined solutions selected in step (b)(vi):

(i) randomly generating a new solution which is separated from the recombined solution by less than a predetermined search radius;

(ii) calculating a coverage level of the new solution; and

(iii) reiterating steps (c)(i) and (c)(ii) until a globally optimized solution is reached.

4. The method of claim 1, wherein step (b) comprises using the genetic algorithm to optimize the number of cameras determined in step (a).

5. The method of claim 4, wherein step (b) comprises:

(i) determining a plurality of random initial solutions, each using the number of cameras determined in step (a);

(ii) performing a local search around each of the random initial solutions to find a locally optimized solution;

(iii) applying a random mutation to each of the locally optimized solutions to obtain a mutated solution, the random mutation comprising a random mutation in the number of cameras in each of the locally optimized solutions;

(iv) recombining the mutated solutions to obtain recombined solutions;

(v) sorting the recombined solutions by coverage level; and

(vi) selecting a number of the recombined solutions having the highest coverage levels for the simulated annealing algorithm.

6. The method of claim 5, wherein step (b)(iv) comprises, when two of the mutated solutions having different numbers of cameras are to be recombined, randomly generating a new number of cameras which is between the different numbers of cameras of the two mutated solutions to be recombined.

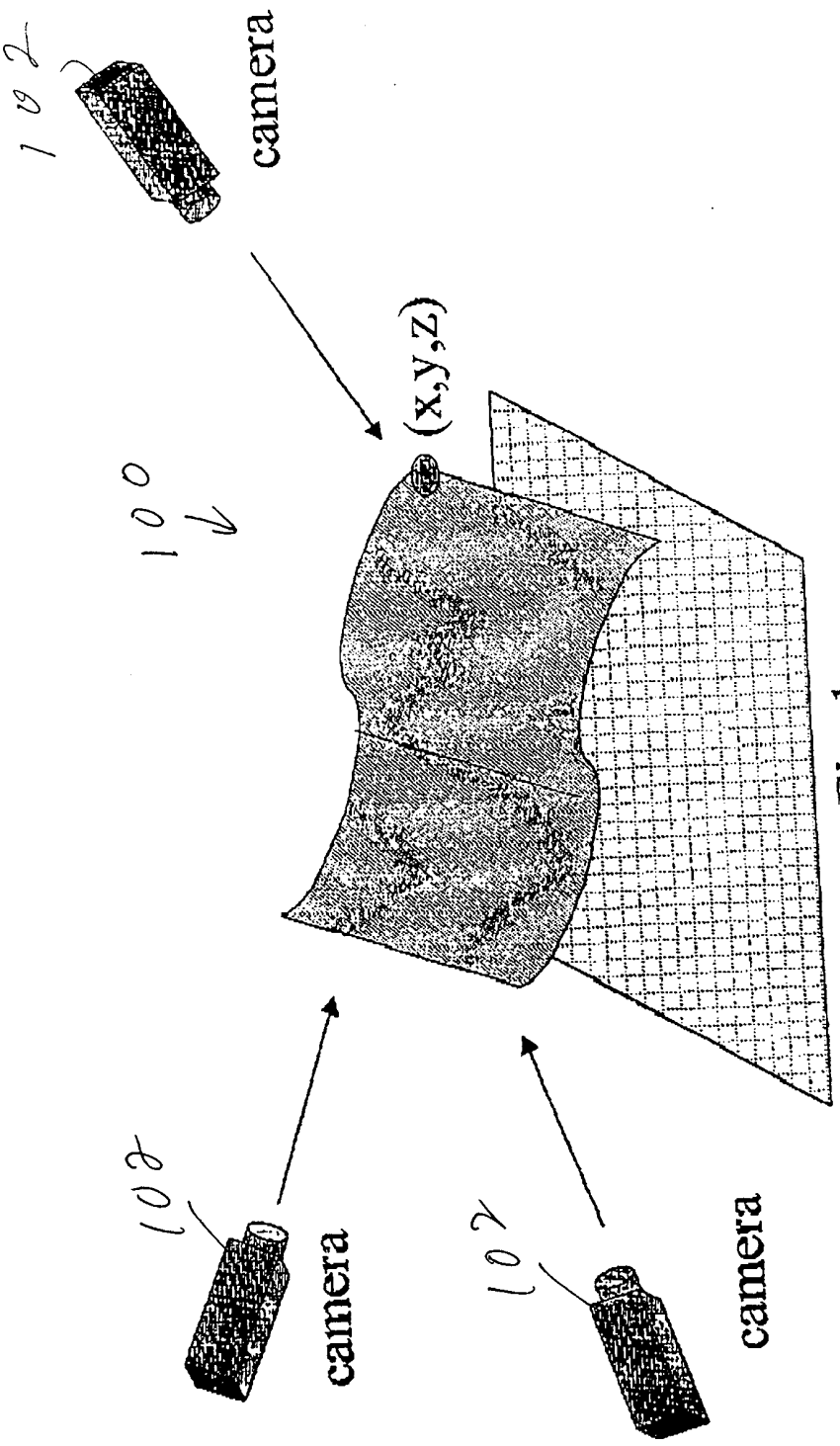


Fig. 1

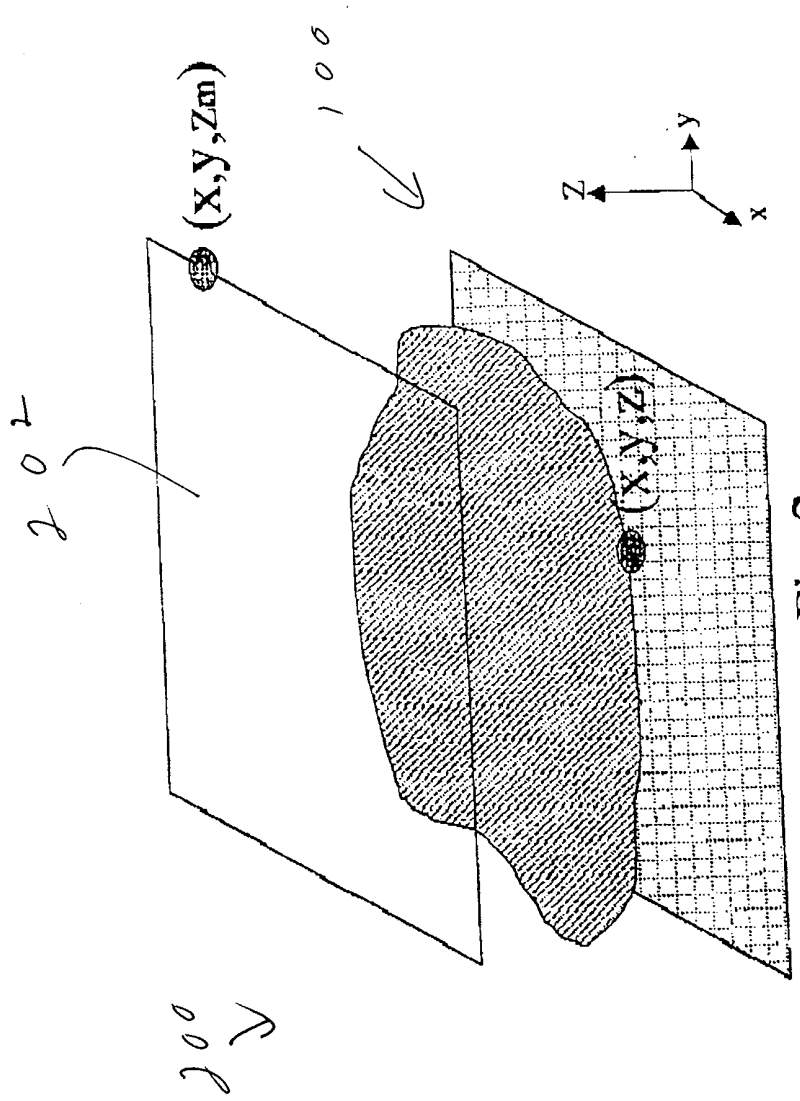


Fig. 2

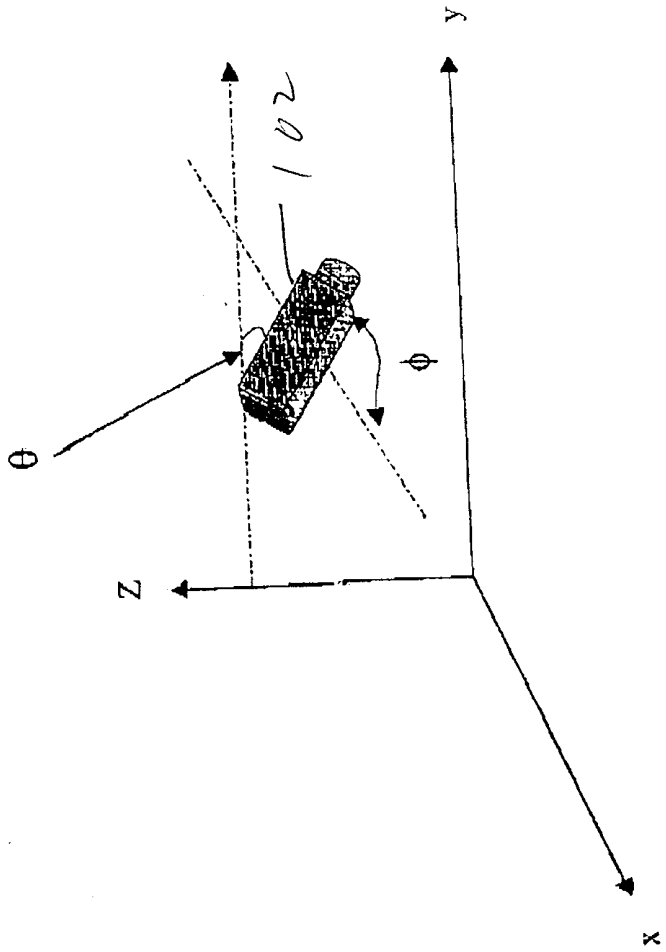


Fig. 3

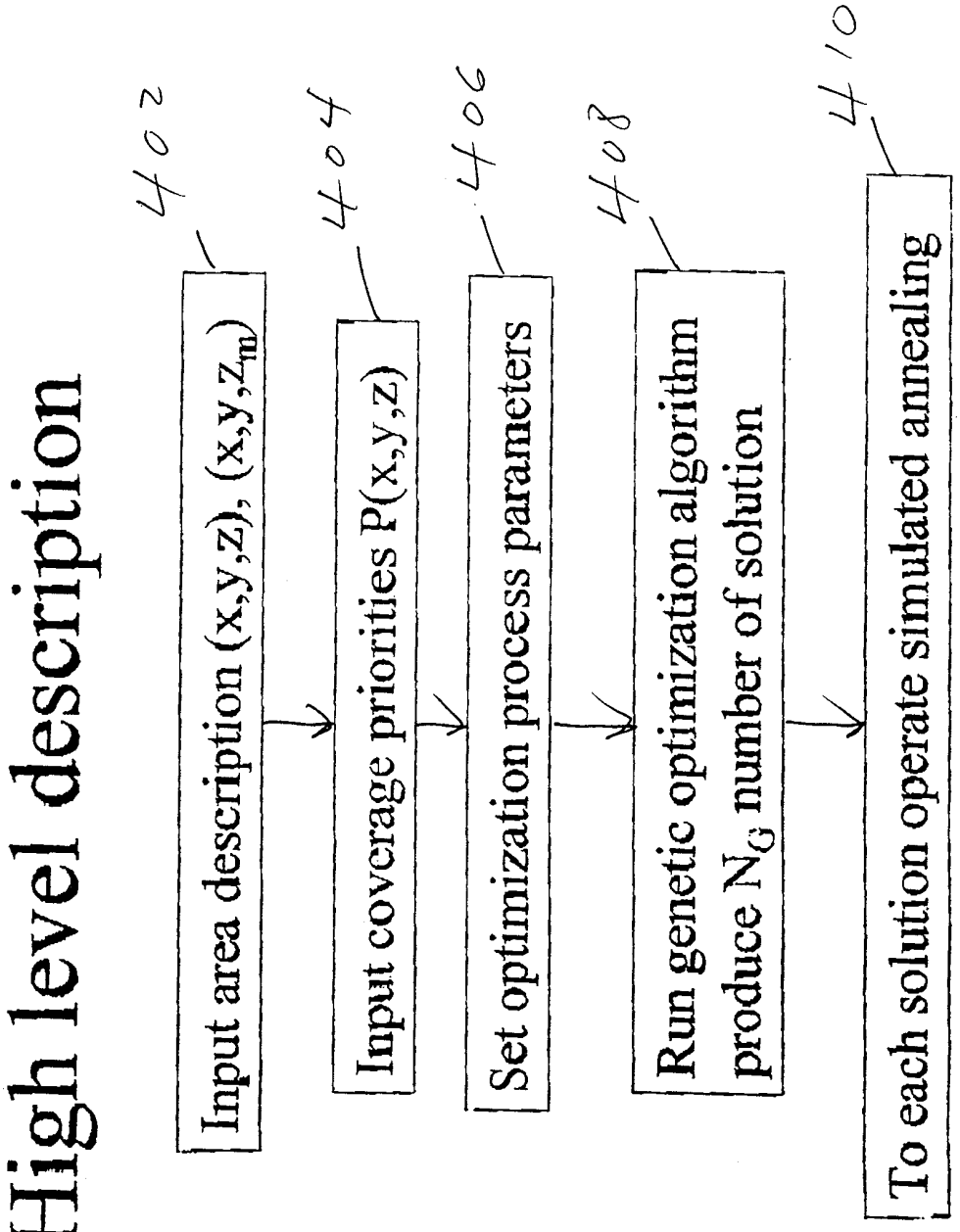

High level description
402
Input area description (x,y,z), (x,y,z_m)
404
Input coverage priorities P(x,y,z)
406
Set optimization process parameters
408
Run genetic optimization algorithm produce N_G number of solution
410
To each solution operate simulated annealing

Fig. 4

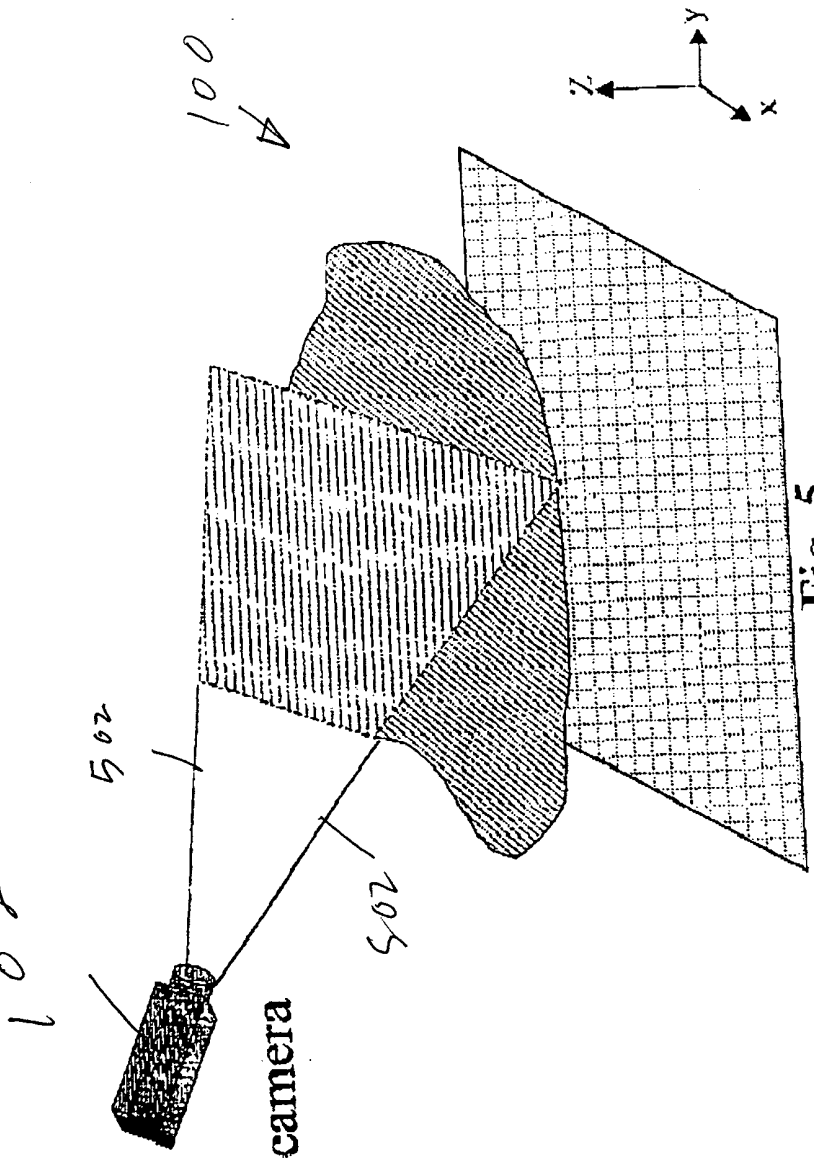


Fig. 5

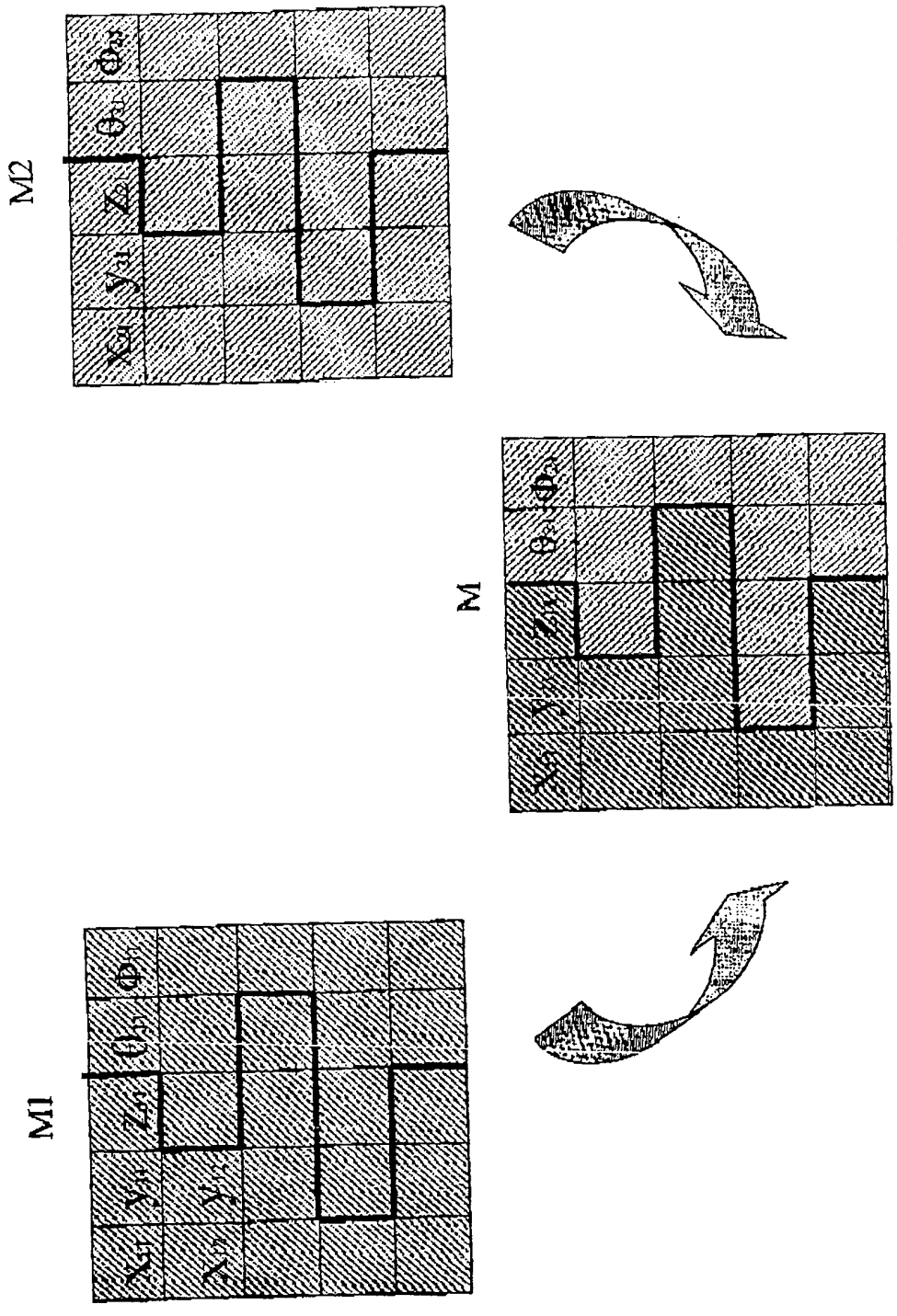


Fig. 6