UNIVERSITÄT AUGSBURG
Universitätsbibliothek

OPEN ACCESS
OPUS AUGSBURG

Approximating Optimal Visual Sensor Placement

Eva Hörster, Rainer Lienhart

Angaben zur Veröffentlichung / Publication details:

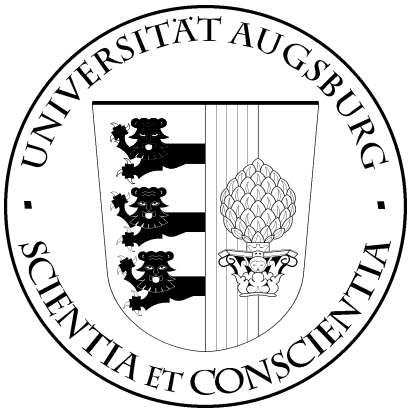
Hörster, Eva, and Rainer Lienhart. 2006. “Approximating Optimal Visual Sensor Placement.” Augsburg: Universität Augsburg.

Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:
Deutsches Urheberrecht
Weitere Informationen finden Sie unter: / For more information see:
https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/

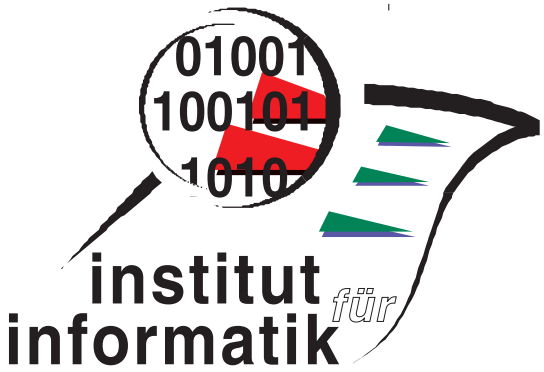
Universität Augsburg



Approximating Optimal Visual Sensor Placement

E. Hörster, R. Lienhart

Report 2006-01 Januar 2006



Institut für Informatik
D-86135 Augsburg

Copyright © E. Hörster, R. Lienhart
Institut für Informatik
Universität Augsburg
D–86135 Augsburg, Germany
http://www.Informatik.Uni-Augsburg.DE
— all rights reserved —

APPROXIMATING OPTIMAL VISUAL SENSOR PLACEMENT

E. Hörster

Multimedia Computing Lab
University of Augsburg
Augsburg, Germany

R. Lienhart

Multimedia Computing Lab
University of Augsburg
Augsburg, Germany

ABSTRACT

Many novel multimedia applications use visual sensor arrays. In this paper we address the problem of optimally placing multiple visual sensors in a given space. Our linear programming approach determines the minimum number of cameras needed to cover the space completely at a given sampling frequency. Simultaneously it determines the optimal positions and poses of the visual sensors. We also show how to account for visual sensors with different properties and costs if more than one kind are available, and report performance results.

1. INTRODUCTION

Visual sensor arrays are used in many novel multimedia applications such as video surveillance, sensing rooms, or smart conference rooms. An important issue in designing sensor arrays is the appropriate placement of the visual sensors such that they achieve a predefined goal. Our goal it to get complete coverage of a given space at a predefined 'sampling rate' guaranteeing that an object in the space will be imaged at a minimum resolution (see Section 2 for a precise definition). Currently designers of multi-camera systems place cameras by hand as there exists no theoretical research on planning visual sensor placement. As video sensor arrays are getting larger, efficient camera placement strategies need to be developed.

Often several different types of cameras are available. They differ in the range of views, intrinsic parameters, image sensor resolutions, optics, and costs. Minimizing the cost of a visual sensor array while maintaining the required resolution (i.e., minimal 'sampling frequency') is also an important issue we consider. Fig. 1 shows one ineffective setup that not even achieves coverage.

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 a distance r (the sensing range of the sensor) [1, 2, 3]. Our work is based on those approaches, but differs in the sensor model, since cameras do not posses circular sensing ranges.

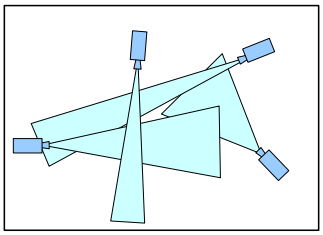


Fig. 1. Example of an inefficient setup we desire to optimize

The sensor placement problem is also closely related to the guard placement problem (AGP) – the problem of determining the minimum number of guards required to cover the interior of an art gallery. It is addressed by the art gallery theorem [4]. Our camera placement problem differs from the AGP in two ways: (1)in the restriction of the field-of-view of cameras in our sensor model due to resolution and sensor properties; (2) in considering cameras with different fields-of-views at different levels of costs. In AGP all guards are assumed to have similar capabilities.

The paper is organized as follows. In Section 2 the problem is formulated. Our approach is presented in Section 3, before Section 4 gives implementation details and reports results. Section 5 concludes the paper.

2. PROBLEM STATEMENT

Definitions: In the following the term $space$ denotes a physical 2D or 3D room, which we want to cover by our visual sensors. Currently, we assume that the room is rectangular.

$Coverage$ means that every point of a given space is sensed with a specified minimal resolution. In this work, the minimal resolution is satisfied if a given point in space is imaged by at least one pixel of a camera that does not aggregate more than $x\ cm^2$ of a surface parallel to the imaging plane through that point. x is expressed in terms of the sampling frequency f_s and converted into the field-of-view of a camera. The field-of-view is defined as a the area in which a pixel aggregates no more that $\frac{1}{f_s} cm^2$ of a surface parallel to the imaging plane. Thus an object that appears in the camera's field-of-view is imaged with at least this resolution assuming the object has

a planar surface orthogonal to the optical axis[1]. Occlusions are not considered. To simplify the derivation we consider only the 2D problem in this paper; however, the presented approach can be extended easily to the third dimension.

Problem Statement: Given a space to be covered at a sampling frequency f_s by visual sensors, we are interested in the following two camera placement problems:

- Determine the minimum number of visual sensors of a certain type as well as their positions and poses in the space such that coverage is achieved.
- Given different types of visual sensors determine how to obtain coverage while minimizing the total cost of the sensor array.

2.1. Modeling a camera's field-of-view

The field-of-view of a camera can be described by a triangle as shown in Fig. 2. The parameters of this triangle can be easily calculated given the (intrinsic) camera parameters and the sampling frequency f_s using well known geometric relations. Defining the field-of-view by a triangle enables us to describe

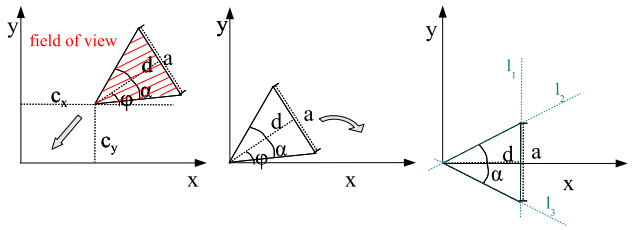


Fig. 2. Deriving the model of a camera's field-of-view

the area covered by a camera at position (c_x, c_y) and pose (φ) linearly. Therefore a camera's field-of-view is first translated to the origin of the coordinate system (Fig. 2 left):

$$x' = x - c_x, \quad y' = y - c_y \tag{1}$$

Then we rotate the field-of-view, so that the optical axis becomes parallel to the x-axis (Fig. 2 center):

$$x'' = \cos(\varphi) \cdot x' + \sin(\varphi) \cdot y' \tag{2}$$

$$y'' = -\sin(\varphi) \cdot x' + \cos(\varphi) \cdot y' \tag{3}$$

The resulting area covered by the triangle (Fig. 2 right) can now be described by three line equations l_1, l_2, l_3:

$$l1: \quad x'' \leq d \tag{4}$$

$$l2: \quad y'' \leq \frac{a}{2d} \cdot x'' \tag{5}$$

$$l3: \quad y'' \geq -\frac{a}{2d} \cdot x'' \tag{6}$$

[1]Clearly the resolution is smaller if the surface is not orthogonal.

Thus by substitution the following three equations define the area covered by the field-of-view of a certain camera:

$$\cos(\varphi) \cdot (x - c_x) + \sin(\varphi) \cdot (y - c_y) \leq d \tag{7}$$

$$-\sin(\varphi) \cdot (x - c_x) + \cos(\varphi) \cdot (y - c_y) \leq \frac{a}{2d} \cdot (\cos(\varphi) \cdot (x - c_x) + \sin(\varphi) \cdot (y - c_y)) \tag{8}$$

$$-\sin(\varphi) \cdot (x - c_x) + \cos(\varphi) \cdot (y - c_y) \geq -\frac{a}{2d} \cdot (\cos(\varphi) \cdot (x - c_x) + \sin(\varphi) \cdot (y - c_y)) \tag{9}$$

2.2. Modeling space

In the ideal case cameras can be placed continuously in the space, i.e. the variables c_x, c_y and φ that define a camera's position and pose are continuous variables. As we are not able to solve our problem for the continuous case we approximate the space by a two-dimensional grid of points. The minimum distance Δ between two grid points in the x- and y-direction is determined by the *spatial sampling frequency* f_a: $\Delta = 1/f_a$. Cameras can only be placed at these discrete grid points, and coverage is ensured only for these grid points. Thus our problem turns into a grid coverage problem. For $f_a \to \infty$ our approximated solution converges to the continuous-case solution.

We only consider rectangular spaces of width w and height h with no obstacles in the room constricting the field-of-view of our visual sensors. Remarks about more complex spaces are made in Section 3.1.

3. LINEAR PROGRAMMING

Considering only one type of cameras, i.e., only cameras with the same field-of-view, we formulate our camera placement problem in terms of minimizing the number of cameras needed subject to the coverage constraint. We assume that our space consists of s_x and s_y grid points in the $x-$ and $y-$ dimension respectively[2]. Visual sensors locations are restricted to these grid points. Similarily we discretize the angle φ defining a camera's pose to s_φ different poses only. A camera at position (c_x, c_y) with orientation φ covers a grid point (x, y) if and only if Eq. 7 to 9 are satisfied.

Thus, we can state the optimization problem as follows:
Problem 1: Given a set of grid points and a camera model, minimize the total number of cameras (by optimally assigning cameras to grid points and angels) while ensuring that every grid point is covered by at least one camera.
In the following we derive an integer programming (ILP) model

[2]Given a rectangular space s_x and s_y can be easily calculated given the room's dimensions and the spatial sampling rate f_a.

to solve this visual sensor placement problem. Our approach is based on the algorithm presented in [2]. Let a binary variable $x_{ij\varphi}$ be defined by:

$$x_{ij\varphi} = \begin{cases} 1 & \text{if a camera is placed at grid point } (i,j) \\ & \text{with orientation } \varphi \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The total number of cameras N is then given by

$$N = \sum_{\varphi=0}^{s_\varphi - 1} \sum_{i=0}^{s_x - 1} \sum_{j=0}^{s_y - 1} x_{ij\varphi} \tag{11}$$

Furthermore we define a binary variable $c(i1, j1, \varphi 1, i2, j2)$:

$$c(i1, j1, \varphi 1, i2, j2) = \begin{cases} 1 & \text{if a camera placed at grid point} \\ & (i1, j1) \text{ with orientation } \varphi 1 \\ & \text{covers grid point } (i2, j2) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$c(i1, j1, \varphi 1, i2, j2)$ can be calculated in advance and stored in a table. Our sensor deployment problem can now be formulated as an IPL model:

$$\min \sum_{\varphi=0}^{s_\varphi - 1} \sum_{i=0}^{s_x - 1} \sum_{j=0}^{s_y - 1} x_{ij\varphi} \tag{13}$$

subject to

$$\sum_{\varphi 1=0}^{s_\varphi - 1} \sum_{i1=0}^{s_x - 1} \sum_{j1=0}^{s_y - 1} x_{i1,j1,\varphi 1} \cdot c(i1, j1, \varphi 1, i2, j2) \geq 1 \tag{14}$$

$$0 \leq i2 \leq (s_x - 1), \quad 0 \leq j2 \leq (s_y - 1)$$

The constraints represented by Eq. 14 ensure that each grid point is covered by at least one camera. The minimum number of cameras that should cover each grid point can be easily changed to e.g. two or more cameras.
To ensure that only one camera is located at each grid point, we can add the constraints:

$$\sum_{\varphi=0}^{s_\varphi - 1} x_{ij\varphi} \leq 1 \tag{15}$$

$$0 \leq i \leq (s_x - 1), \quad 0 \leq j \leq (s_y - 1)$$

The number of variables $x_{ij\varphi}$ in our ILP is $s_x \cdot s_y \cdot s_\varphi$. Thus, if we increase the number of grid points to achieve a better approximation of the continuous case, the number of variables and constraints in our IPL increases accordingly.

3.1. Different types of cameras

A very similar problem arises if several types of cameras with different sensor resolutions and optics (i.e. focal lengths) are available. Then we have for each type of camera k different field-of-view parameters d_k and a_k (see Fig. 2) and a cost K_k. Our objective is to find the configuration of cameras that minimizes the total cost of the visual sensors while ensuring coverage. The optimization problem is formulated as follows:
Problem 2: Given a set of grid points and k types of visual sensors with costs K_k and field-of-view parameters d_k, a_k minimize the total cost of the sensor array (by optimally assigning cameras to grid points and angles) while ensuring coverage for every grid point.
To solve this problem we need to modify the previous solution. We define a binary variable $x_{ij\varphi k}$ as follows:

$$x_{ij\varphi k} = \begin{cases} 1 & \text{if a camera of type } k \text{ is placed at} \\ & \text{grid point } (i,j) \text{ with orientation } \varphi \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

The total cost C of the visual sensors is then calculated by:

$$C = \sum_{k=1}^{k} K_k \Big(\sum_{\varphi=0}^{s_\varphi - 1} \sum_{i=0}^{s_x - 1} \sum_{j=0}^{s_y - 1} x_{ij\varphi k} \Big) \tag{17}$$

Similar to Eq. 12 a binary variable $c^k(i1, j1, \varphi 1, i2, j2)$ is defined for each camera type k by:

$$c^k(i1, j1, \varphi 1, i2, j)) = \begin{cases} 1 & \text{if a camera of type } k \text{ with} \\ & \text{orientation } \varphi 1 \text{ is placed at} \\ & \text{grid point } (i1, j1) \text{ and covers} \\ & \text{grid point } (i2, j2) \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

Again the value of all variables $c^k(i1, j1, \varphi 1, i2, j2)$ can be calculated in advance. Like in *Problem 1* the solution to our visual sensor placement problem is found by solving the following ILP model:
Minimize the cost function

$$C = \sum_{k=1}^{k} K_k \Big(\sum_{\varphi=0}^{s_\varphi - 1} \sum_{i=0}^{s_x - 1} \sum_{j=0}^{s_y - 1} x_{ij\varphi k} \Big) \tag{19}$$

subject to

$$\sum_{k1=1}^{k} \sum_{\varphi 1=0}^{s_\varphi - 1} \sum_{i1=0}^{s_x - 1} \sum_{j1=0}^{s_y - 1} x_{i1,j1,\varphi 1,k1} \cdot c^k(i1, j1, \varphi 1, k1, i2, j2) \geq 1 \tag{20}$$

$$0 \leq i2 \leq (s_x - 1), \quad 0 \leq j2 \leq (s_y - 1)$$

To ensure that at each grid point only one camera is located we need to add the constraints:

$$\sum_{k=1}^{k} \sum_{\varphi=0}^{s_\varphi - 1} x_{ij\varphi k} \leq 1 \tag{21}$$

$$0 \leq i \leq (s_x - 1), \quad 0 \leq j \leq (s_y - 1)$$

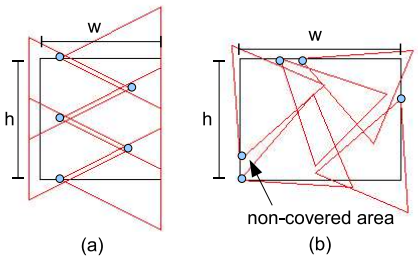


Fig. 3. Results of the ILP minimization for identical cameras of two sample configuratations (a) and (b)

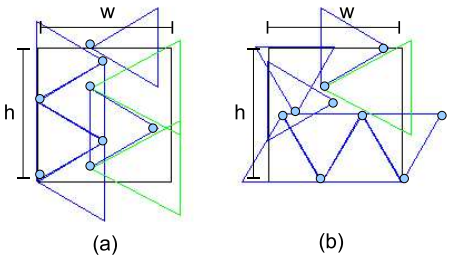


Fig. 4. Results of minimizing the total costs of the sensor arrays for two different parameters settings

The number of variables $x_{ij\varphi k}$ in our ILP is $k \cdot s_x \cdot s_y \cdot s_\varphi$, i.e. the number of variables increases with the number of grid points and available camera types as do the constraints.

More complex spaces: We have only considered rectangular spaces so far. More complex rooms can be easily included as linear constraints into our framework as long as they are convex. Additionally, the ILPs' extension to the third dimension is straightforward, but has been excluded in the discussion due to space limitations.

4. EXPERIMENTAL RESULTS

The ILPs have been implemented in C++ using the $lpsolve$ package [5]. First we evaluated our proposed solution to *Problem 1*: Given a single camera type minimize the total number of cameras subject to the coverage constraints. Fig. 4 shows two results with two sample configurations. Blue dots mark camera locations; red lines mark the field-of-views of the cameras. The sampling frequency f_s and the camera type were set equally in both experiments, the other parameters were set as follows:

- Experiment (a): room: 120×120, $f_a = \frac{1}{10}$, $s_\varphi = 2$
- Experiment (b): room: 160×120, $f_a = \frac{1}{20}$, $s_\varphi = 16$

It can be observed in Fig. 4(b) that some areas are uncovered. This is due to the low spatial sampling frequency f_a. Coverage is only assured for discrete grid points and the distance between two grid points depends on f_a. To ensure near complete coverage, f_a needs to be increased.

We also evaluated the cost minimization ILP for two camera types with different field-of-views. A cost of \$70 was assumed for the cameras with the larger field-of-view, while only \$50 were assumed for the cameras with the smaller field-of-views. Fig. 4 shows two results. Blue dots denote camera locations, while blue and green lines mark the field-of-views of the two different camera types.

The size of the room, the sampling and spatial sampling frequency f_s and f_a were constant for both experiments. In the first and second experiment we chose $s_\varphi = 2$ (Fig. 4(a)) and $s_\varphi = 8$ (Fig. 4(a)), respectively. The total cost of the sensor arrays in Fig. 4(a) was \$440 and \$420 in Fig. 4(b).
The above presented ILP problems are practically solvable for only a small number of grid points. For a large number Chakrabarty et al. [2] propose a divide-and-conquer algorithm.

5. CONCLUSION AND FUTURE WORK

We have presented an approach to optimally place visual sensors in a given space such that coverage is achieved for discrete grid points. We have proposed two different algorithms. The first one minimizes the number of cameras covering a sensing field at a given sampling frequency. The second algorithm minimizes the total cost of the sensor array while covering a space with sensors of different properties.
Future work will include a slack variable for coverage, i.e., only a predefined percentage of grid points needs to be covered. Additionally it has to be investigated how to handle a large number of grid points.

6. REFERENCES

[1] S. Sahni and X. Xu, "Algorithms for wireless sensor networks," *Intl. Journal on Distributed Sensor Networks*, vol. 1(1), pp. 35–56, 2005.

[2] H. Qi K. Chakrabarty, S. S. Iyengar and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Transaction on Computers*, vol. 51(12), pp. 1448–1453, 2002.

[3] J. Wang and N. Zhong, "Efficient point coverage in wireless sensor networks," *Journal of Combinatorial Optimization*, to appear.

[4] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, New York, 1987.

[5] P. Notebaert M. Berkelaar and K. Eikland, "lpsolve: Open souce (mixed-integer) linear programming system," *Eindhoven U. of Technology*, http://groups.yahoo.com/group/lp_solve/files/Version5.5/.