# Optimal Camera Placement

*Avital Steinitz*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-69
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-69.html

May 9, 2012

Acknowledgement

This work was made possible by NSF grant 100265 from The University of Texas in Dallas and by army grant 554968 from The University of Pennsylvania.

# Optimal Camera Placement

by

Avital Avigad Steinitz

A thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science
in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Ruzena Bajcsy, Chair
Professor Stuart J. Russell
Professor Michael Lustig

Spring 2012

Optimal Camera Placement

# Abstract

Optimal Camera Placement

by

Avital Avigad Steinitz

Master of Science in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Ruzena Bajcsy, Chair

Optimal Camera Placement is a task shared by applications such as Next Best View planning and Camera Network planning, and it is used in exploration of unknown environments, surveillance cameras placement and scene reconstruction. More specifically, Optimal Camera Placement consists of evaluating a visibility objective that depends on the scene and the camera placement and optimizing it with respect to to the camera variable, where different applications give rise to different visibility objectives.

Despite its wide applicability, many of the application listed above are limited in scale, efficiency and the complexity of the scene and sensor model they can handle. This limitation stems primarily from the common practice to discretize the space of camera configurations, thus viewing the problem as a combinatorial optimization problem whose size grows exponentially with the number of cameras and which is known to be NP-hard. Another drawback of the "early-discretization" approach is its inherent inaccuracy with respect to objectives that are continuous by nature, such as visible surface area.

In this work we propose an expressive formalization of the visibility objective which is amenable to continuous optimization techniques. The formalism we propose is generic and may be used for many of the existing applications; it is inherently more accurate than the discrete approaches as it views both the objects' and the cameras' configuration spaces as continuous and it can handle complex scenes with occlusions and camera placement constraints as well as sophisticated camera models. We describe an algorithm for evaluating the proposed objective and its gradient, and present simulation results showing the quality of the obtained camera placements and their efficient computation.

# Contents

# Acknowledgements

I am grateful to Ruzena Bajcsy for accepting me to her multidisciplinary research group and for assigning me to work with Ram Vasudevan. I could not possibly express all my gratitude to Ram, who has been a friend and a mentor. Our numerous arguments and conversation about philosophy, science and life literally kept me going.

Orna Kupferman from my home university, The Hebrew University of Jerusalem, has taught with patience and kindness how to write and much more. Miki Lustig and Tommy Kaplan have been personal and professional role models.

Being a Graduate Teaching Assistant under the instruction of Stuart Russell was one of the most educating and satisfying experiences I had at Berkeley.

I would like to thank my dear friends Daniel Aranki, Michael Mrejen and Baruch Sterin here at Berkeley, Elad Eban and David Katz oversees, and my dear friend Roi Reichart from the right coast.

Finally, I would like to thank my supporting and loving family, Naomi, Yuval, Ruth, Dan, Haggai, Tal, Sivan, Dror-Ron and the whole new Roi-Ariel.

This work was made possible by NSF grant 100265 from The University of Texas in Dallas and by army grant 554968 from The University of Pennsylvania.

# Chapter 1

# Introduction

Optimal Camera Placement is a task shared by many applications involving sensor placement. We begin our discussion of Optimal Camera Placement by describing compelling motivating examples from the existing literature.

*Next Best View* is a sequential planning task, where in each step a new measurement of the environment is obtained and the location of the next measurement is determined based on all previous measurements. Normally, the location of the next measurement is chosen to maximize an objective that represents the visibility of the scene, often in the form of the (negative) uncertainty about the objects in the scene. In a recent work, Nüchter et al. [2003] leverage planar techniques for inferring an Optimal Camera Placement from analyzing planar slices of the three dimensional space.

*Camera Network Control* refers to simultaneous placement of multiple cameras. In another recent work, Schwager et al. [2011a] place cameras mounted on quad-rotors to obtain optimal coverage of a planar polygonal shape. As in the Next Best View setting, the cameras location are chosen to maximize an objective that intuitively captures how well the target polygon is being viewed by the cameras.

Lee [1991] discusses placement of fire-towers to monitor and prevent fire. Bodor et al. [2007] consider optimal camera placement specifically for surveillance tasks by finding an optimal camera placement for observing a set of well trodden paths taken by pedestrians. Olague and Mohr [2002] show how to place multiple cameras in order to minimize scene reconstruction error. Last, in a technical and insightful application, Cowan and Modayur [1993] show how to place a camera and an illumination source in order to enhance edge detectability.

All the aforementioned applications share a common step in which a *known* scene $S$ is associated with a visibility objective

$$Q_S : \mathbf{C} \to \mathbf{R} \tag{1.1}$$

that quantifies how well the scene $S$ is viewed from a camera placement $C \in \mathbf{C}$, which is then optimized with respect to $C$

$$\max_{C \in \mathbf{C}} \; Q_S(C) \; . \tag{1.2}$$

This step is the Optimal Camera Placement step, and in this work we propose accurate, efficient and generic means for its solution.

It is important to emphasize that also exploratory tasks, where the scene and the observed targets are *unknowns*, utilize this formulation of finding Optimal Camera Placement in a *known* environment (that is precisely the setting in the Next Best View task). The reason being that in every step in the exploratory process the location of the next measurement is chosen *based on the previously collected measurements*, that is, based on the current model of the scene. Therefore, by restricting our focus to finding optimal camera placement in known environments we do not make unreasonable assumptions.

We now give a high-level description of the common visibility objective found in existing literature. Most works begin by specifying their scene modeling, usually consisting of polygonal objects, and by specifying a *local visibility objective*

$$q : (x, C) \mapsto \mathbf{R} \tag{1.3}$$

that quantifies how well a single feature, $x$, is viewed from a camera placement $C$. The feature $x$ may be a point, an edge, a facet or the like, and the camera placement $C$ may specify the camera configuration, possibly consisting of location, orientation, focal length, etc. In fact, $C$ may consist of the camera specification for multiple cameras as well as the locations and orientations of the sources of illumination. For ease of terminology, we refer to $C$ by the term camera placement also when there is more than one camera and when its configuration consists also of other parameters besides its geometric location (such as focal length).

The local visibility objective evaluates *how well* a feature is viewed, possibly taking into account any properties of the sensor model as well as the application at hand. An example for encapsulating the sensor model in the local visibility objective would be assigning poor visibility to features residing outside the camera's field of view, and an example for encapsulating the specific application in the local visibility objective would be assigning high visibility only to features that are viewed by more than one camera (which is necessary for scene reconstruction). In many of the previous works about Optimal Camera Placement a feature $x$ is a single point, the camera placement $C$ consists of specifying the location (and perhaps also the orientation) of a single camera, and the local visibility objective $q$ is the foreshortened length of the feature on the image plane, given by $1 - \cos(\alpha)$ where $\alpha$ is the angle between the line of sight from $C$ to $x$ and the normal to the object surface at $x$.

Next the *global visibility objective* is formalized as

$$Q_S(C) = \sum_i \chi(x_i, C) \cdot q(x_i, C) \tag{1.4}$$

where $x_1, \ldots, x_n$ is a finite set of features, and $\chi$ is a *visibility indicator* which specifies whether a feature $x$ is visible from $C$:

$$\chi(x, C) = \begin{cases} 1 & x \text{ is visible from } C \\ 0 & x \text{ is occluded} \end{cases} \tag{1.5}$$

Most often, the values of the visibility indicator for all features and for a finite selection of possible camera placements are computed and stored in a matrix called *the viewability matrix*. Then based on the information stored in the viewability matrix, a combinatorial optimization algorithm is used to pick the optimal camera placement (see Tarbox and Gottschlich [1995] and Pito [1999] for a discussion about the computation of the viewability matrix).

This approach has three major limitations:

1. A discrete set of features is inherently inaccurate in capturing continuous objectives such as visible surface area.
2. The computation of the viewability matrix is time consuming and its complexity grows quickly with the granularity of the discretization of the camera placement space and with the number of features.
3. This formalism discards the spatial relationships among the features and is oblivious to which features are near each other and which are distant. This conceptual loss of information manifests itself by casting the problem into a combinatorial optimization form, namely the Art-Gallery problem. The Art-Gallery problem has been studied extensively and shown to be NP-hard to solve and in some cases even to approximate (see O'Rourke [1987] and Eidenbenz et al. [1998]). Roughly speaking, being NP-hard implies that the Art-Gallery problem scales exponentially with the size of the problem. In our case the size of the problem is the size of the sensor model times the number of cameras (also the accuracy parameter, which determines the number of possible camera placements and the number of features, influences the complexity of the problem, only through the base of the exponent). Therefore, the NP-hardness implies that large problems, problems that require high accuracy or multiple cameras placement, would be practically infeasible. In view of this latter discovery, we can revisit the applications listed above and observe that, indeed, all of them involved only a small number of cameras and a relatively small model.

In this work we propose an alternative generic formalism for evaluating visibility. More technically, we substitute the sum over a finite set of features by an integral over the entire scene. Formally, we consider the visibility objective of the form

$$Q(C) = \int ||\dot{x}(t)|| \cdot \chi(x(t), C) \cdot q(x(t), C) \partial t \tag{1.6}$$

where $x : [0, 1] \to \mathbf{R}^3$ is a smooth parametrization of the target object in the scene and $\chi$ and $q$ are the same as before. Before we proceed by describing the advantages of the proposed formulation, we wish to note that we do not view the assumption that the parametrization of the scene is smooth as a limitation because of a few reasons: Primarily, the parametrization of compact and opaque real world shapes are in fact smooth, closed and continuous. Additionally, a twice differentiable model is no less realistic then a piecewise linear model or any other parametrized model. Furthermore, the family of twice-differential paths is dense in the space of continuous paths and therefore any two dimensional object (including piecewise linear objects) can be approximated using a twice differentiable path to any arbitrary precision. Last, from a more applicative perspective, the scene models are often generated from a set of discrete measurements through curve fitting, and requiring that the fitted model be smooth poses no procedural limitation.

The primary motivation for proposing this objective is the observation that the *local visibility of a single feature* (or a finite set of features) is a discontinuous and non-differentiable function of the camera placement, whereas the *global visibility of the entire object* (roughly corresponding to the visible surface area) is a continuous and differentiable function of the camera placement. Therefore, being differentiable our objective lends itself to continuous optimization techniques, thus it can be optimized efficiently and is scalable, yielding an efficient optimal camera placement algorithm.

To summarize, the main contributions of this work are:

- The Optimal Camera Placement formalism we propose is generic in the sense that it allows for constrains on the camera placement and for other shapes besides polygons.
- The Optimal Camera Placement formalism we propose enables optimizing over the visibility rather than considering it as a constraint.
- The objective of the Optimal Camera Placement formalism we propose is almost everywhere differentiable and can be optimized efficiently on each one of its domains.

We briefly elaborate on the last item, namely the domains of the gradient of the objective. In general, the gradient of the objective is not well-defined everywhere. To see this consider for example the piecewise linear object illustrated in Figure 1.1 and the surface-area global visibility objective (corresponding to a constant local visibility objective). In this case the global visibility objective would be a piecewise constant function and its gradient would be defined on the domains $A$, $B$, $C$, $D$, $E$ and $F$. In this work we show how to optimize the global visibility objective within each one of its domains. Furthermore, we discuss in detail a special case of great interest, the commonly used foreshortened length local visibility objective, for which the global visibility objective *is* always differentiable.

There are three limitations of our work that require clarification:

1. Optimal Camera Placement problem is inherently non-convex, as can be easily seen from observing symmetric scenes (and see Schwager et al. [2011b] for a rigorous discussion). Therefore, gradient based approaches for finding optimal camera placement are bound to end up finding local and not necessarily global optima. However discouraging, an interesting property of our objective is that also its local optima seem reasonable to a human observer, even if not globally optimal.

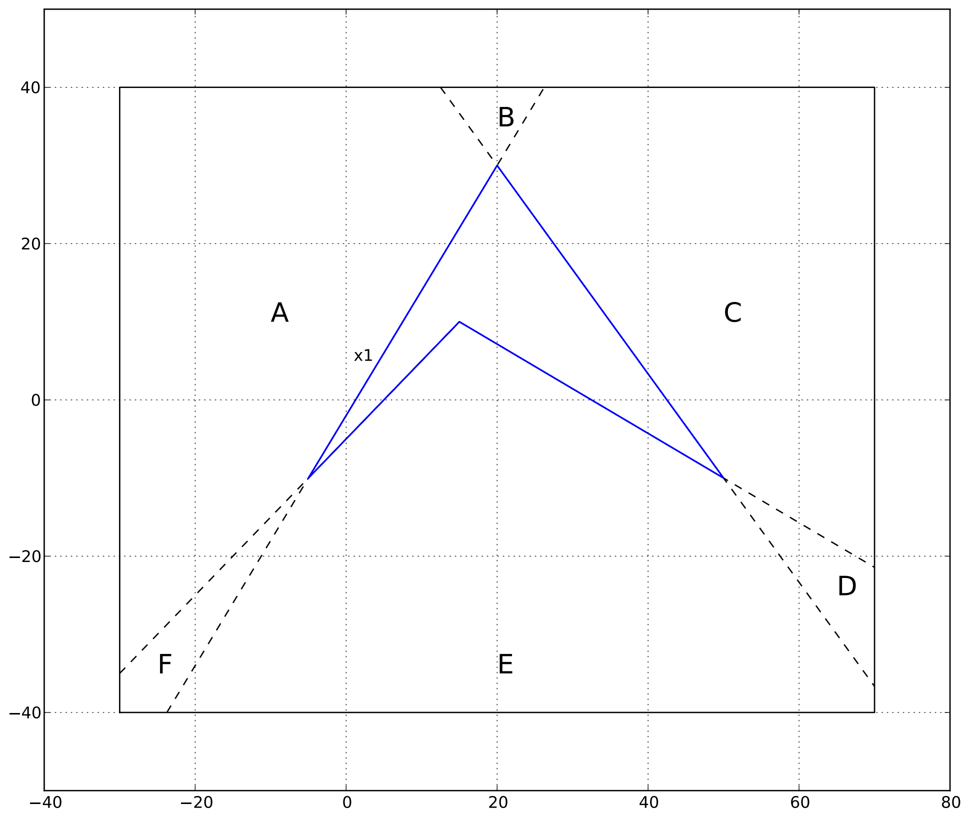

Figure 1.1. A piecewise linear object ($x_1$) and the domains of the gradient of the surface-area global visibility objective ($A,\ B,\ C,\ D,\ E,\ F$).

2. Currently, a stable evaluation of the objective and its gradient requires assuming some knowledge about the scene, namely a bound on the number of inflection points in the parametrization of the scene. In Chapter 6 we propose an alternative formulation that would be free of this assumption and that seems to be a promising area for future research.

3. In this work we handle simple camera placement space. In particular, we only consider the geometric placement of the camera, and we do not discuss illumination and other optical parameters. However, our findings remain valid for any local visibility objective that is sufficiently smooth with respect to its arguments, therefore our results extend to more expressive cameras configurations optimization.

We conclude this chapter by briefly reviewing possible extensions and augmentations of our model. As we mentioned in the previous paragraph, we wish to consider a variant of our objective that would be differentiable everywhere and would therefore enable more efficient optimization. We discuss one such variant in Chapter 6. Another interesting research problem would be studying how well our model captures human experts' notion of Optimal Camera placement. To this end a local visibility objective could be learned from and compared to experts' Optimal Camera Placement. We would also be interested in considering our approach in various more demanding contexts. Specifically, when implemented as a part of an exploratory Next Best View system one has to study the overall stability and convergence properties of our model, a task which becomes even more interesting when the observed scene is dynamic. Additionally, an implementation for a cameras network should take into account communication considerations (as discussed in Schwager et al. [2011a]), which brings up the question how our Optimal Camera Placement algorithm would operate under communication uncertainties. Last, a more long term goal would be to extend these ideas to other sensors and perhaps develop a theory of optimal sensor placement for multiple sensor types.

# Chapter 2

# Assumptions and Notations

We start by describing our model for objects, scenes and cameras. The current model is two dimensional but it could be naturally generalized to the three dimensional case.

A *scene* is a finite collection of objects $x_1, x_2, \ldots, x_m$ and a camera placement constraint given by the level-set of a differentiable function $g$. We now give a detailed formal description of the objects and constraints we consider.

## 2.1 Modeling Scenes

We consider objects that are parametrized as twice-differentiable closed paths in $\mathbf{R}^2$. Formally, an *object* $x$ is a function

$$x : S^1 \to \mathbf{R}^2 \; . \tag{2.1}$$

We use Newton's dot notation for the first and second derivatives with respect to the parametrization variable $t$:

$$\dot{x}(t) = \frac{\partial x}{\partial t}(t), \qquad \ddot{x}(t) = \frac{\partial^2 x}{\partial t^2}(t) \; . \tag{2.2}$$

We find it natural to refer to the parametrization variable in temporal terms and to say *$x$ at time $t$* to refer to $x(t)$.

Also, relying on geometric intuition, we say *the tangent to $x$ at time $t$* to refer to $\dot{x}(t)$.

We denote the normal to the surface of the object by $n$. The normal at time $t$ is obtained by a simple rotation of the tangent at time $t$:

$$n(t) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot \dot{x}(t) \tag{2.3}$$

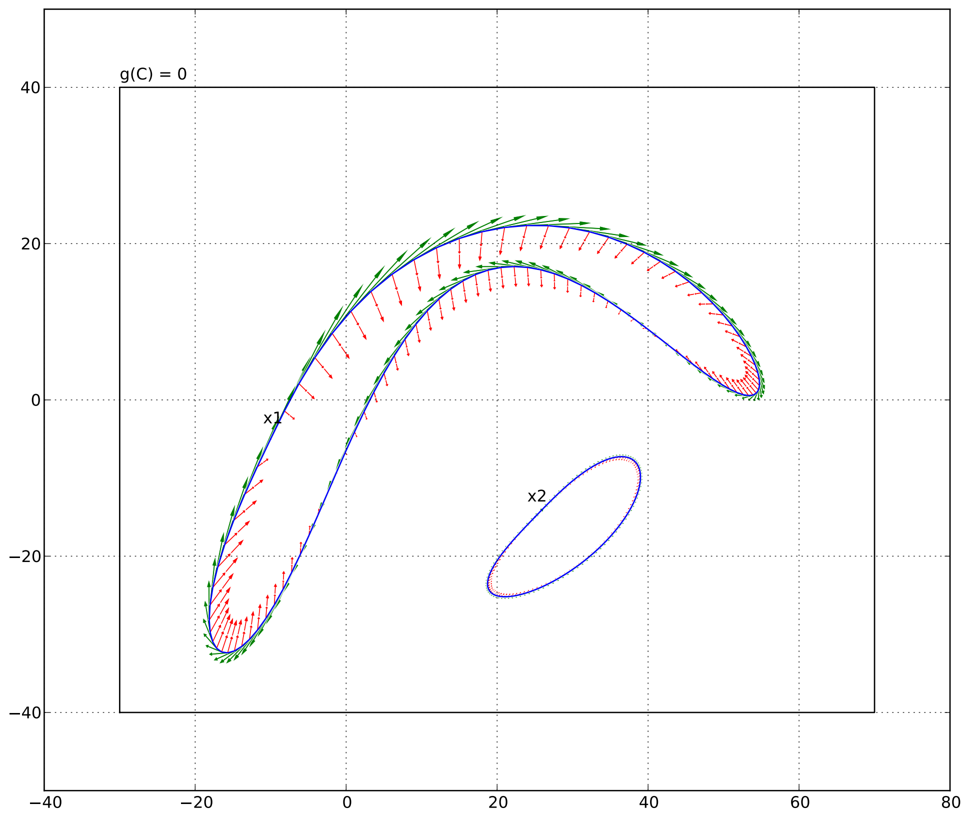

Figure 2.1. An illustration of a scene and its components.

Figure 2.1 illustrates a scene and its components. The scene in the figure consists of two objects, $x_1$ and $x_2$, marked in blue. The thicker green arrows mark the tangents to the objects, $\dot{x}_1$ and $\dot{x}_2$, and the thinner double red arrows mark the second derivatives of the paths parametrizations, $\ddot{x}_1$ and $\ddot{x}_2$. It is interesting to note that the second derivatives of $x_1$, which is non-convex, shift from pointing inwards to pointing outwards, and that there are times where the second derivative is aligned with the first derivative, those times are the *inflexion times*, which will later play a significant role with respect to the evaluation of the gradient of our visibility objective.

## 2.2 Modeling Cameras Placement

A camera placement $C$ may specify the full camera configuration, possibly consisting of location, orientation, focal length, etc. Tarabanis et al. [1995] give a comprehensive list of camera configuration parameters. In this work, however, we handle very simple an intuitive camera placement space, consisting only of location.

Formally, we denote the space of cameras configurations $\mathbf{C}$, and a camera placement is an element

$$C \in \mathbf{C} \; . \tag{2.4}$$

In our case, $\mathbf{C}$ is $\mathbf{R}^2$ for a single camera placement, and $\mathbf{R}^{2k}$ for $k$ cameras placement.

We enable constraining the the camera placement space to the 0 level-set of a differentiable constraint function

$$g : \mathbf{C} \rightarrow \mathbf{R} \; . \tag{2.5}$$

Figure 2.1 also illustrates a camera placement constraint marked as a black rectangle.

# Chapter 3

# Problem Statement

In this chapter we formally define the *global visibility objective* that captures the notion of how well a scene is viewed given a specific camera placement. We begin by discussing the simple case of a single camera placement in a scene consisting of a single object and proceed with the more general case of multiple objects and multiple cameras.

Intuitively, we quantify the visibility of an object $x$ from a camera location $C$ in a sense that generalizes the notion of visible surface area. The visible surface area of an object can be computed as the length of the visible portions of the object, or equivalently, the integral over the norm of the gradient. We generalize this notion by allowing for more general integrands, where different integrands correspond to different features of interest and different applications.

## 3.1 The Visibility Indicator

### 3.1.1 Single Object Scenes

Consider a scene $S$ consisting of a single object $x$. We say that *$x$ is occluded from $C$ at time $t$* if "there is something in the way between $x(t)$ and $C$". In other words, $x(t)$ is occluded from $C$ if there is a point $x(t')$, for some $t' \neq t$, which is occluding (or "hiding") the point $x(t)$ from $C$.

Formally, let $f_{\text{closer}}(t', t)$ be an indicator that $x(t')$ is closer to $C$ than $x(t)$

$$f_{\text{closer}}(t', t) = \begin{cases} 1 & ||x(t) - C|| > ||x(t') - C|| \\ 0 & \text{else} \end{cases} \tag{3.1}$$

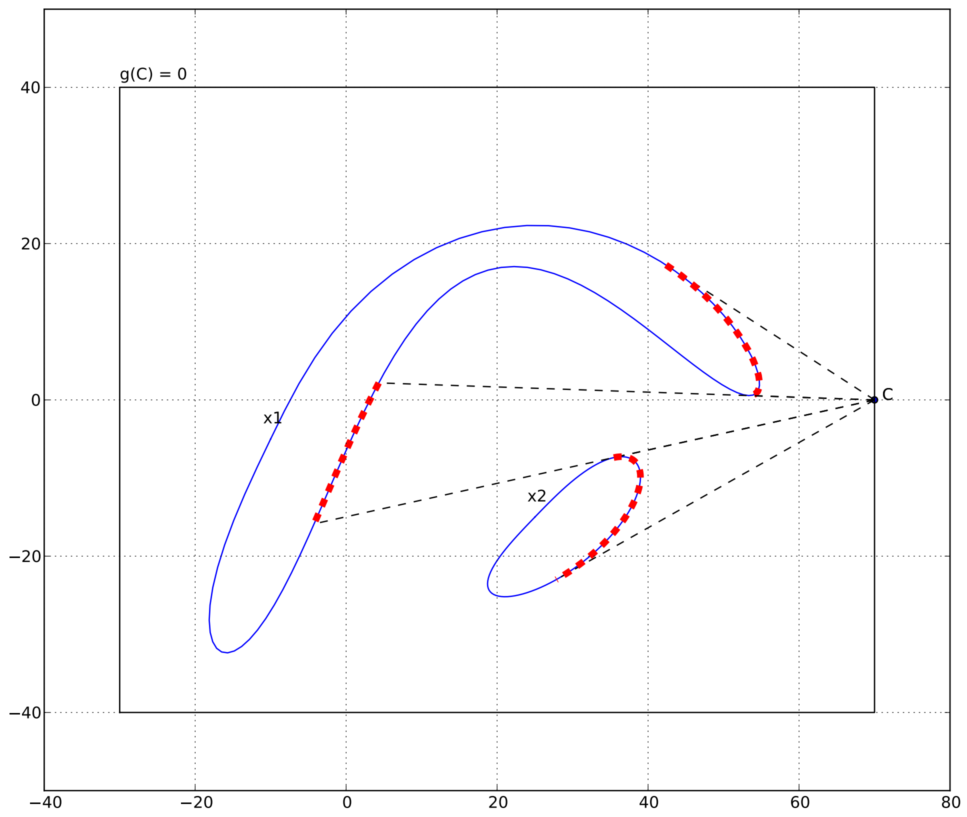

Figure 3.1. The visible portions of in a two objects scene.

and $f_{\text{aligned}}(t', t) = 1$ be an indicator that $x(t'), x(t)$ and $C$ are co-linear

$$f_{\text{aligned}}(t', t) = \begin{cases} 1 & (x(t) - C)^{\intercal} \cdot (x(t') - C) = ||x(t) - C|| \cdot ||x(t') - C|| \\ 0 & \text{else} \end{cases} \tag{3.2}$$

where $^{\intercal}$ is the notation we use for transpose. Next, we say that the point $x(t)$ is occluded from $C$ by $x(t')$ (or equivalently $x(t')$ occludes $x(t)$ from $C$) if

$$f_{\text{closer}}(t', t) \cdot f_{\text{aligned}}(t', t) = 1 \ . \tag{3.3}$$

Equivalently, we write $\mathbf{1}_{condition}$ to denote the *condition*'s indicator function that evaluates to 1 whenever *condition* holds true, and write the expression in Equation 3.3 as

$$\mathbf{1}_{||x(t)-C||-||x(t')-C||>0} \cdot \mathbf{1}_{(x(t)-C)^{\intercal} \cdot (x(t')-C)=||x(t)-C|| \cdot ||x(t')-C||} = 1 \ . \tag{3.4}$$

Given the single object scene $S$, we define the visibility indicator

$$\chi_S : [0,1] \times \mathbf{C} \to \{0,1\} \tag{3.5}$$

$$\chi_S(t,C) = \begin{cases} 0 & \text{The point } x(t) \text{ is occluded from } C \text{ by some } x(t') \\ 1 & \text{else} \end{cases} . \tag{3.6}$$

Formally, we have

$$\chi_S(t,C) = 1 - \max_{t' \in S^1 \setminus t} \left\{ \mathbf{1}_{||x(t)-C|| > ||x(t')-C||} \cdot \mathbf{1}_{(x(t)-C)^\intercal \cdot (x(t')-C) = ||x(t)-C|| \cdot ||x(t')-C||} \right\} . \tag{3.7}$$

### 3.1.2 Multiple Objects Scene

We extend our definition of visibility indicator to scenes with multiple objects in the natural way, taking into account inter-objects occlusions. Consider a scene $S$ consisting of the objects $x_1, x_2, \ldots, x_m$.

We say that $x_i(t)$ is occluded from $C$ by $x_j(t')$ (or equivalently $x_j(t')$ occludes $x_i(t)$ from $C$) if

$$\mathbf{1}_{||x_i(t)-C|| - ||x_j(t')-C|| > 0} \cdot \mathbf{1}_{(x_i(t)-C)^\intercal \cdot (x_j(t')-C) = ||x_i(t)-C|| \cdot ||x_j(t')-C||} = 1 \ . \tag{3.8}$$

We define the $i$'th visibility indicator as

$$\begin{aligned} \chi_S^i(t,C) = 1 - max \Bigg\{ & \max_{t' \in S^1 \setminus t} \left\{ \mathbf{1}_{||x_i(t)-C|| > ||x_i(t')-C||} \cdot \mathbf{1}_{\langle x_i(t)-C, x_i(t')-C \rangle = ||x_i(t)-C|| \cdot ||x_i(t')-C||} \right\}, \\ & \max_{j \in [m] \setminus i} \left\{ \max_{t' \in S^1} \left\{ \mathbf{1}_{||x_i(t)-C|| > ||x_j(t')-C||} \cdot \mathbf{1}_{\langle x_i(t)-C, x_j(t')-C \rangle = ||x_i(t)-C|| \cdot ||x_j(t')-C||} \right\} \right\} \Bigg\} . \end{aligned} \tag{3.9}$$

Figure 3.1 illustrates a two-objects scene and the visible portions with respect to a single camera placement.

## 3.2 Local Visibility Objective

We now discuss the notions of *local* and *global* visibility objectives which generalize the notion of visible surface area. Intuitively, the local visibility objective quantifies *how well* a point $x(t)$ is viewed from a cameras placement $C$, and the global visibility objective is the integral of the local visibility objective over the visible portions of the scene.

Consider a scene $S$ consisting of the objects $x_1, x_2, \ldots, x_m$. We associate a continuous differentiable function with every object in the scene $S$

$$q_S^i : (t, C_1, \ldots, C_l) \mapsto \mathbf{R} \tag{3.10}$$

where $q_S^i(t, C_1, \ldots, C_l)$ quantifies how well the point $x_i(t)$ is viewed from $C_1, \ldots, C_l$ (regardless of whether or not it is visible or occluded). Depending on the application, the local visibility objective may depend on the placement of single or multiple cameras. For example, in the context of accurate scene reconstruction the local visibility objective may take as input the configurations of two or more camera placements, whereas in the context of standard robotics tasks, where the foreshortened length is a popular measure for quality of feature visibility, the local visibility objective may take as input the placement of a single camera. Additionally, as mentioned in the introduction, the local visibility objective may also encapsulate parts of the sensor model such as the field of view specification.

We proceed by offering three examples of useful local visibility objectives.

### 3.2.1 Visible Surface Area

In the case where the application is to maximize the visible surface or scene reconstruction, any visible point is viewed equally well and therefore the local visibility objective is a constant

$$q_S^i(t, C) = 1 \tag{3.11}$$

and its gradient is 0.

### 3.2.2 Foreshortened Length

In many robotics and vision applications, a point is considered to be viewed well if the line of sight is perpendicular to the tangent to the surface. Figure 3.2 illustrates the angle between the line of sight and the surface normal.

This notion of local visibility objective is formalized by

$$q_S^i(t, C) = 1 - \frac{n_i(t)^\intercal \cdot (x_i(t) - C)}{||n_i(t)|| \cdot ||x_i(t) - C||} \tag{3.12}$$

In practice, to make this objective differentiable, we consider a slight variant of it

$$q_S^i(t, C) = 1 - \frac{\left(n_i(t)^\intercal \cdot (x_i(t) - C)\right)^2}{||n_i(t)||^2 \cdot ||x_i(t) - C||^2} \tag{3.13}$$

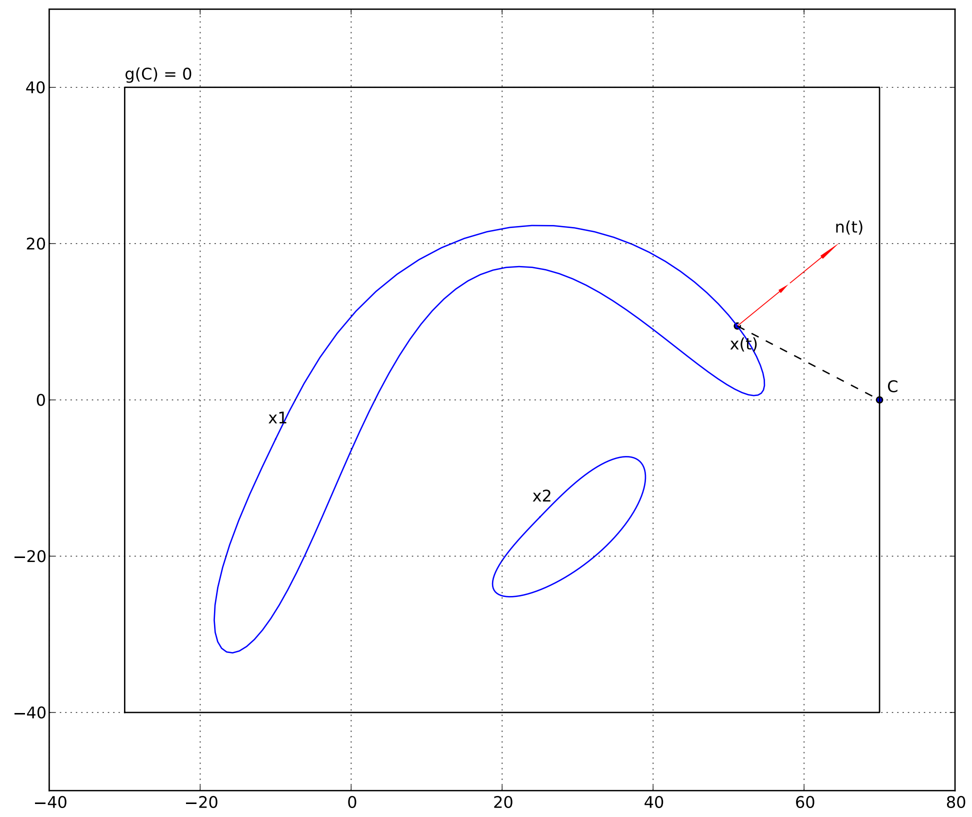

Figure 3.2. The angle between the line of sight and the surface normal.

### 3.2.3 Reconstruction Error

We propose another interesting local visibility objective that quantifies the reconstruction error. Assuming a sensor model with finite angular accuracy $\delta$, a feature that is determined by two cameras may reside in any point inside a polygon that is bounded by the rays surrounding the lines of site with angle $\delta$.

Figure 3.3 illustrates the reconstruction error polygon.

## 3.3 Global Visibility Objective

The *global visibility objective* that corresponds to the scene $S$ and to the local visibility objective $q$ is the integral over the product of the local visibility objective and the visibility

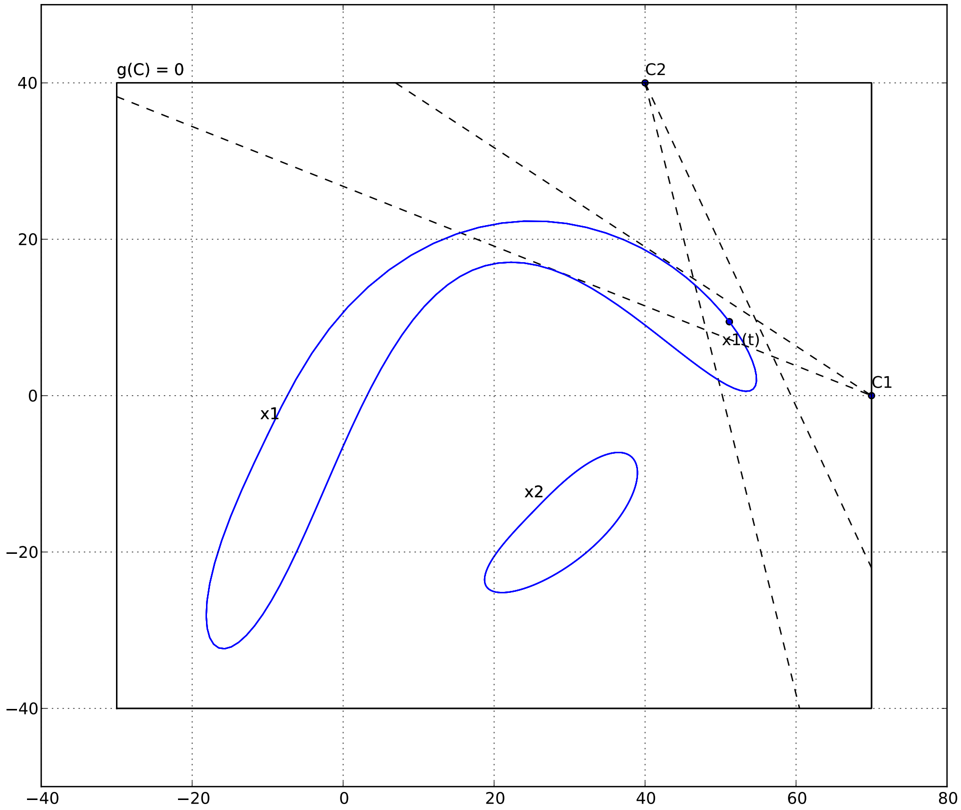

Figure 3.3. The reconstruction error.

indicator. Formally, for the case of a single camera and a scene consisting of a single object, we have

$$Q_S(C) = \int_0^1 ||\dot{x}(t)|| \cdot \chi(t, C) q(t, C) \partial t \ . \tag{3.14}$$

This formalism naturally extends to scenes consisting of multiple objects by summation over the objects in the scene, and to multiple cameras by maximization of the integrand with respect to the cameras. Formally, given a scene $S$ consisting of $m$ objects and their corresponding local visibility objectives, the global visibility objective is defined as

$$Q_S(\vec{C}) = \sum_{i=1}^{m} \int_0^1 \max_{j_1,\ldots,j_l} \left[ q_S^i(t, C_{j_1}, \ldots, C_{j_l}) \prod_{p=1}^{l} \chi_S^i(t, C_{j_p}) \right] \partial t \ . \tag{3.15}$$

## 3.4 The Optimization Problem

Given a scene $S$ and its corresponding global visibility objective and a camera placement constraint $q$, the Optimal Camera Placement problem is the optimization problem

$$\begin{aligned} &\max_{\vec{C}} \ Q_S(\vec{C}) \\ &\text{subject to: } g(\vec{C}) = 0 \end{aligned} \tag{3.16}$$

We note that whenever there is no ambiguity we shall simplify the notation and suppress the super and sub indexing.

# Chapter 4

# Previous Work

There are numerous works about Optimal Camera Placement dating back to the late eighties. In this chapter we present the various approaches to solving the problem by discussing a selection of representative works. There is more than one way to lay down the taxonomy of the works in the field. Traditionally, the works are surveyed by the vision task they aim to solve or the a priori knowledge about the task they assume (see for example Maver and Bajcsy [1993]), and then by their procedural approach and by the specific features and constraints they consider (see for example the survey by Tarabanis et al. [1995]).

To capture best the contribution of this work we review the objectives used in those previous works, as well as their evaluation and optimization. More specifically, we focus on the evaluation of the visibility of a single point, the form of the global visibility objective and the scene model.

The evaluation of visibility of a single point is done manually, algorithmically or analytically, where by algorithmically we mean that the visibility is determined by a procedure that does not necessarily correspond to an analytic expression. We make the distinction between the analytic and the algorithmic approach because only the analytic approach would later fit into the optimization framework.

The form of the global visibility objective is most often a sum over the local visibility values corresponding to a finite set of features. We make the distinction between the *finite sum approach* and the *integration approach* because in the presence of occlusions only the latter could fit into the continuous optimization framework. Furthermore, as discussed in the introduction, we find the integration approach also more accurate and appropriate for many applications.

Last, in most works the scene models are limited to piecewise linear shapes (polyhedra) whereas in our work we merely require that the shapes be sufficiently smooth.

## 4.1 The Finite Sum Approach

### 4.1.1 Manual Visibility

In Bodor et al. [2007] the authors consider surveillance tasks. The objects they consider consist of a single straight line each, and the occlusion relations between them are assumed to be known.

Olague and Mohr [2002] consider scene reconstruction error minimization. They mention that not all feature points are visible from all camera placements, but they do not discuss the process of determining which points ar visible and which aren't.

### 4.1.2 Algorithmic Visibility

Tarbox and Gottschlich [1995] give a detailed description of an algorithm for determining the visibility of a set of points from a set of possible camera placements. Then, they use the foreshortened local visibility objective to construct a so called *viewability matrix*, whose rows correspond to a single camera placement and whose columns correspond to a feature point. They reduce the problem of finding the optimal camera placement to the problem of finding a subset of the rows that would give the optimal visibility of all the features. To this end, they experiment with three combinatorial optimization schemes and report their results.

Bouyagoub et al. [2010] follow a similar path of work, and they too construct the viewability matrix (though without specifying exactly how) and experiment with various combinatorial optimization schemes for finding the optimal set of rows.

Chen and Davis [2008] also relate to determining static occlusions by checking whether something stands "in the way" between the camera and the feature of interest.

Pito [1999] uses ray tracing techniques to scan and model objects "on the fly", and using a combinatorial optimization heuristic to pick a next best views sequence that would eventually cover the entire surface of the target object. He also discusses in detail the pros and the cons of discretizing the camera placement space.

### 4.1.3 Analytic Visibility

Maver and Bajcsy [1993] consider the equations for the straight line between a feature and a camera and check whether it intersects the scene model. Next, they pick the next best view as the point that is exposed to as much of the occluded region as possible. To this end, they associate every camera placement with a histogram representing the currently occluded regions that it observes and the next best view is the location with the highest scoring histogram.

Another two lines of work using the analytic synthesis approach are Cowan and Kovesi [1988], Cowan [1988] and Cowan and Bergman [1989], and Tarabanis and Tsai [1991], Tarabanis et al. [1996] and Tarabanis et al. [1991]. These works demonstrate use of analytic methods for finding sensor and light source configurations that guarantee meeting multiple features detection constraints: visibility, focus, field of view, resolution and illumination. These works model analytically the regions that would be occluded by a convex polygon or polyhedron (non-convex bodies must be broken down to convex components for this method to be applicable) which enables them to solve a system of linear equations in order to find a point in space from which no feature is occluded. We note that this work is suitable for finding perfect visibility of all the features, but it is not suitable for optimizing over the visibility itself. Furthermore, this work relies heavily on modeling the scene using polyhedra, which hinges on the accuracy of the model.

On a side note, an original instance of their ideas, still within the same analytic scope, is the work of Cowan and Modayur [1993] in which the illumination source is located to enable edge detection by shading.

## 4.2 The Integration Approach

Perhaps the most similar work to ours is Schwager et al. [2011b]. There, the visibility is evaluated as the integral of a local “information per pixel” objective over the entire scene, and it is optimized using gradient descent. Notably, their objective extends naturally and efficiently to multiple cameras.

The three major factors in which this work differs from ours are that we consider the visibility as a part of the optimization objective rather than as a constraint, thus we can handle cases where the entire scene cannot possibly be observed. We also allow for more flexible scenes with occlusions and for constraints on the camera placement space.

In Fleishman et al. [2000] the objective captures a discretized version of the visible surface area of multiple polygonal objects. Furthermore, occlusions are taken into account, as well as constraints on the possible camera placement and multiple cameras configurations. The means for determining which points are visible from where are analytic. However, Fleishman et al. [2000] consider only a very limited discrete setting, with discrete local visibility objective, discrete shapes (polygons) and discrete space of possible cameras placements. Then, they solve the resulting discrete optimization problem using a greedy algorithm.

We restate that we allow for more general scene modeling and the local visibility objective, and that our optimization techniques are more efficient and scalable.

## 4.3 Non-Geometric and Approaches and Sub-Modular Optimization

Krause and Guestrin [2007] discusses optimal sensor placement in a probabilistic setting, for which sub-modular optimization is used. Their work differs from our in many respects. First, our work would be applicable also for optimization over functions that are not sub-modular (for example, visibility by multiple cameras for reconstruction). Second our modeling of visibility is deterministic and geometric property rather than a probabilistic property, which is a more appropriate model for many phenomena. Last, the sub-modular optimization algorithm is generally less accurate and efficient than continuous optimization.

# Chapter 5

# Optimal Camera Placement

In this chapter we present our algorithm as well as some simulation results. We present the algorithm through an example, which naturally leads to the full specification of the algorithm as well as its underlying assumptions.

## 5.1 Example

Recall that our goal is to solve the optimization problem

$$\begin{aligned} &\max_{\vec{C}} \; Q(\vec{C}) \\ &\text{subject to: } g(\vec{C}) = 0 \end{aligned} \tag{5.1}$$

where

$$Q(C) = \int_0^1 ||\dot{x}(t)|| \cdot \chi(t, C) q(t, C) \partial t \; . \tag{5.2}$$

for a given scene $S$, camera placement constraint $g$ and local visibility objective $q$. To simplify notation, we assume throughout this chapter that the the "speed" factor, $||\dot{x}(t)||$, is embedded within the local visibility objective. This gives the simpler formulation

$$Q(C) = \int_0^1 \chi(t, C) q(t, C) \partial t \; . \tag{5.3}$$

We begin by specifying the input to the problem, namely $S$, $q$, and $g$:

### 5.1.1 Example - Input

We consider a scene consisting of a single object, as illustrated in Figure 3.2. The single object in the scene is a non-convex Bezier curve

$$x(t) = \sum_{i=0}^{n} \binom{n}{i} \cdot (1-t)^{n-i} \cdot t^i \cdot P_i \ , \tag{5.4}$$

whose first and second derivatives are

$$\dot{x}(t) = n \cdot \sum_{i=0}^{n-1} \binom{n-1}{i} \cdot (1-t)^{n-i} \cdot t^i \cdot (P_{i+1} - P_i) \tag{5.5}$$

$$\ddot{x}(t) = (n-1) \cdot n \cdot \sum_{i=0}^{n-2} \binom{n-2}{i} \cdot (1-t)^{n-i} \cdot t^i \cdot (P_{i+2} + 2 \cdot P_{i+1} - P_i) \ . \tag{5.6}$$

We chose this special form of curves simply because of its smoothness and non-convexity. To obtain the interesting "banana" shaped curve while maintaining twice differentiability we chose a curve of degree 18 (marked in blue).

We wish to constrain the camera placement to the 100 × 80 rectangle centered at $[0, 20]$ (marked in Figure 3.2 in black). To this end we use a standard approximate rectangular two-dimensional differentiable function whose 0 level-set is the desired constrained camera placement space.

Last, the local visibility objective is the foreshortened length described in equation 3.13.

### 5.1.2 Example - The Global Visibility Objective

Naïvely, we would proceed by taking the derivative of the objective

$$\frac{\partial Q}{\partial C}(C) = \int_0^1 \left[ \frac{\partial \chi}{\partial C}(t, C) \cdot q(t, C) + \chi(t, C) \cdot \frac{\partial q}{\partial C} \right] \partial t \ . \tag{5.7}$$

Unfortunately, that route is blocked because $\frac{\partial \chi}{\partial C}(t, C)$ does not always exist (and when it does exist, it equals 0). We therefore proceed by presenting a reformulation of the global visibility objective.

Consider the camera placement $C = [70, 0]$, and note that the times where $x$ is visible form two contiguous intervals, marked in red in Figure 5.1.

We use the term *occlusion boundaries* to refer to the endpoints of the visibility intervals, $x(t_1), x(t_2), x(t_3)$ and $x(t_4)$. We use the term *occlusion times* to refer to their corresponding times, $t_1,\ t_2,\ t_3$ and $t_4$. Viewing the occlusion times as functions of $C$ enables us to write

$$Q(C) = \int_{t_0(C)}^{t_1(C)} q(t, C) \partial t + \int_{t_2(C)}^{t_3(C)} q(t, C) \partial t \ . \tag{5.8}$$

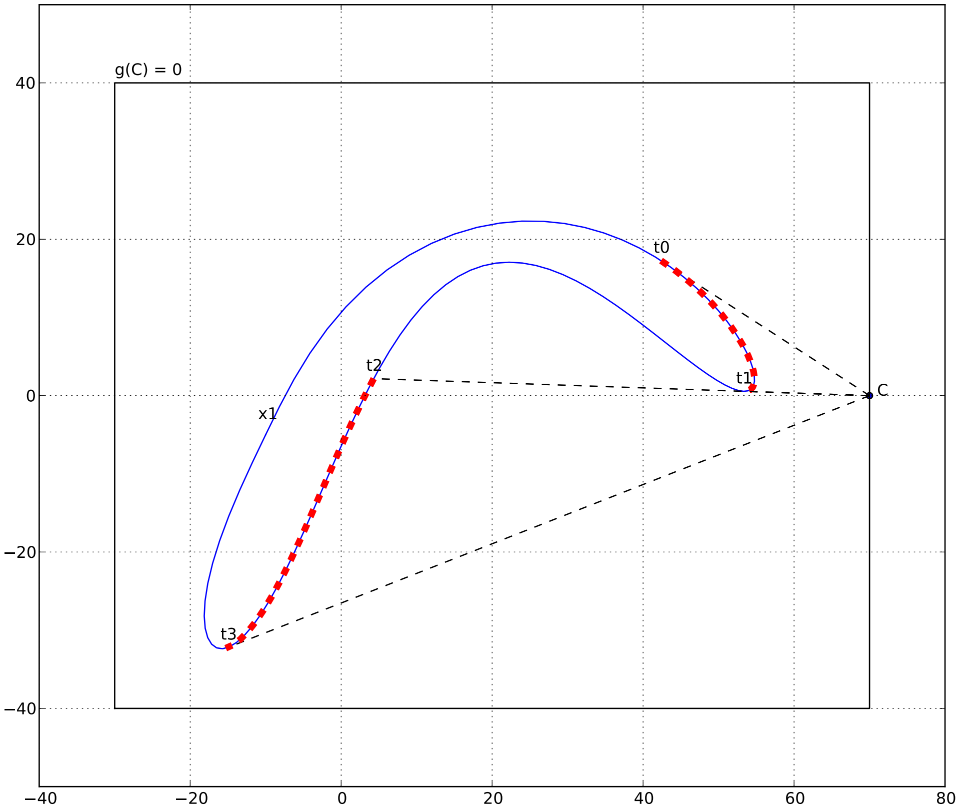

Figure 5.1. The visible intervals corresponding to single camera placement.

Assuming that the occlusion boundaries are continuous functions of the camera placement $C$ we can take the derivative of $Q$ with respect to $C$ using Leibniz rule:

$$\begin{aligned}\frac{\partial Q}{\partial C}(C) = &\left( -\frac{\partial t_0}{\partial C}(C) \cdot q(t_0, C) + \frac{\partial t_1}{\partial C}(C) \cdot q(t_1, C) + \int_{t_0(C)}^{t_1(C)} \frac{\partial q}{\partial C}(t, C) dt \right) + \\ &\left( -\frac{\partial t_2}{\partial C}(C) \cdot q(t_2, C) + \frac{\partial t_3}{\partial C}(C) \cdot q(t_3, C) + \int_{t_2(C)}^{t_3(C)} \frac{\partial q}{\partial C}(t, C) dt \right) \end{aligned} \qquad (5.9)$$

Note that the specific local visibility objective we chose vanishes at $t_0(C), t_1(C)$ and at $t_3(C)$, and therefore we have the simpler expression

$$\frac{\partial Q}{\partial C}(C) = \left( \int_{t_0(C)}^{t_1(C)} \frac{\partial q}{\partial C}(t, C) dt \right) + \left( -\frac{\partial t_2}{\partial C}(C) \cdot q(t_2, C) + \int_{t_2(C)}^{t_3(C)} \frac{\partial q}{\partial C}(t, C) dt \right) \qquad (5.10)$$

We proceed by discussing the derivatives of the occlusion boundary functions with respect to $C$. We write the occlusion boundary as implicit functions of $C$ and take their implicit derivates. We make a distinction between occlusion boundaries where the line of sight is tangent to the surface of the object (such as $t_0, t_1$ and $t_3$) and others. We refer to the former occlusion boundaries as *tangency times* and to the latter occlusion boundaries as *intersection times*. The two sets can be written as

$$T_{||}(C) = \{t : \text{ the line going through } C \text{ and } x(t) \text{ is tangent to } x \text{ at } t\} \tag{5.11}$$

$$T_{\nparallel}(C) = \{t : \text{ the line going through } C \text{ and } x(t_1)\text{, for some } t_1 \in T_1(C)\text{, intersects } x \text{ at } t\} \tag{5.12}$$

where $T_{||}$ is the set of tangency times and $T_{\nparallel}$ is the set of interesection times.

We now proceed to compute the derivative of $T_{\nparallel}$ though its implicit characterization. Recall that two points $u, v \in \mathbf{R}^2$ define the single line connecting them, denoted $L(u, v)$, which is given by the equation

$$L(u,v) \doteq \left\{ w : \left\langle v - w, \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (v-u) \right\rangle = 0 \right\} \ . \tag{5.13}$$

Plugging in $x(T_{||}(C)),\ x(T_{\nparallel}(C))$ and $C$ we write $T_{\nparallel}$ implicitly as

$$\Big(C - x(T_{||}(C))\Big)^{\intercal} \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \Big(C - x(T_{\nparallel}(C))\Big) = 0, \tag{5.14}$$

and taking its implicit derivative we have

$$\begin{aligned} &\left(I - \dot{x}(T_{||}(C)) \cdot \frac{\partial T_{||}}{\partial C}(C)\right)^{\intercal} \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \Big(C - x(T_{\nparallel}(C))\Big) + \\ + \ &\left(I - \dot{x}(T_{\nparallel}(C)) \cdot \frac{\partial T_{\nparallel}}{\partial C}(C)\right)^{\intercal} \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}^{\intercal} \cdot \Big(C - x(T_{||}(C))\Big) = 0 \ . \end{aligned} \tag{5.15}$$

By rearranging we obtain

$$\begin{aligned} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \Big(x(T_{||}(C)) - x(T_{\nparallel}(C))\Big) = &\left(\frac{\partial T}{\partial C}(C)\right)^{\intercal} \cdot \underbrace{\dot{x}(T(C))^{\intercal} \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \Big(C - x(T_{\nparallel}(C))\Big)}_{=0} - \\ &\left(\frac{\partial T_{\nparallel}}{\partial C}(C)\right)^{\intercal} \cdot \dot{x}(T_{\nparallel}(C))^{\intercal} \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \Big(C - x(T_{||}(C))\Big) \end{aligned} \tag{5.16}$$

and finally

$$\frac{\partial T_{\nparallel}}{\partial C}(C) = \frac{1}{\dot{x}(T_{\nparallel}(C))^{\intercal} \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \Big(C - x(T_{||}(C))\Big)} \cdot \Big(x(T_{\nparallel}(C)) - x(T_{||}(C))\Big)^{\intercal} \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} . \tag{5.17}$$

Even though the computation of the derivative is complete, we make a couple of technical substitutions that would make the expression of the gradient more interpretable. Specifically we make the substitutions

$$x(T_{||}(C)) - C \mapsto \frac{\dot{x}(t)}{||\dot{x}(t)||} \cdot ||x(T_{||}(C)) - C|| \tag{5.18}$$

$$x(T_{\nparallel}(C)) - x(T_{||}(C)) \mapsto \frac{\dot{x}(t)}{||\dot{x}(t)||} \cdot ||x(T_{\nparallel}(C)) - x(T_{||}(C))|| \tag{5.19}$$

and write

$$\frac{\partial T_{\nparallel}}{\partial C}(C) = \frac{||x(T_{\nparallel}(C)) - x(T_{||}(C))||}{||C - x(T_{||}(C))||} \cdot \frac{1}{\dot{x}(T_{\nparallel}(C))^{\intercal} \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \dot{x}(T_{||}(C))} \cdot \dot{x}(T_{||}(C))^{\intercal} \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{5.20}$$

We now interpret the expression for the gradient of $T_{\nparallel}$ by making four observation about how we would expect $T_{\nparallel}(C)$ to change as we slightly move $C$ in various directions.

- We expect no change in $T_{\nparallel}(C)$ as we move $C$ in the direction of $\dot{x}(T_{||}(C))$.
- The larger the distance between $C$ and $x(T_{||}(C))$ the *smaller* should be the change in $T_{\nparallel}(C)$.
- The larger the distance between $x(T_{\nparallel}(C))$ and $x(T_{||}(C))$ the *larger* should be the change in $T_{\nparallel}(C)$.
- The more aligned are $\dot{x}(T_{||}(C))$ and $\dot{x}(T_{\nparallel}(C))$ the *larger* should be the change in $T_{\nparallel}(C)$.

These observations account for the terms in equation 5.20.

It is important to note that $\frac{\partial T_{\nparallel}}{\partial C}(C)$ is not defined whenever $\dot{x}(T_{||}(C))$ and $\dot{x}(T_{\nparallel}(C))$ are perfectly aligned. In general, this implies that the global visibility objective is not always differentiable (as in the case of the visible surface area and a scene consisting of polygons). However, in the special case of the foreshortened local visibility objective, the global objective remains differentiable because the local objective vanishes exactly where the gradient of the occlusion times explodes.

We have practically computed all the components in the expression for the gradient of the global visibility objective from Equation 5.10. It remains to use an off-the-shelf non-linear optimization package to find a solution for the optimization problem. We proceed by describing the details of the actual computation of the gradient which is the core of our algorithm.

### 5.1.3 Example - The Algorithm

In this sub-section we describe the details of our algorithm for the special case described in the example. Our algorithm uses a non-linear optimization tool for solving the optimization problem 3.16. The crux of the algorithm is in the computation of the objective and its gradient, which we now describe in detail.

In general, the input to the algorithm is a parametrized scene $S$, a camera placement constraint $g$ and an initial camera placement $C_0$. In what follows we describe the run of algorithm for the scene, camera placement constraint and initial camera placement described earlier in this section and visualized in Figure 5.1.

Given the scene $S$, the local visibility objective $q$ and a camera placement $C$, we evaluate the gradient $\partial Q(C)$ (or the objective $Q(C)$) as follows:

1. We compute $T_{||}(C)$ and $T_{\nparallel}(C)$ and their gradients (when necessary):
   (a) We compute $T_{||}(C)$ by finding the roots of the function
   $$f_{T_{||}}(t) = n(t)^{\intercal} \cdot (x(t) - C) \tag{5.21}$$
   In the case depicted in Figure 5.1 the computation in this step would yield the occlusion times $t_0,\ t_1$ and $t_3$.
   (b) We then compute $T_{\nparallel}(C)$ using a "ray-tracing" procedure. More specifically, for every tangency time $T_1 \in T_{||}(C)$ we find the zeros of the function
   $$f_{T_{\nparallel}}(t) = d(x(t), l(C, x(T_1))) \tag{5.22}$$
   where $l(\mathrm{pt}_1, \mathrm{pt}_2)$ is the straight line going through the points $\mathrm{pt}_1$ and $\mathrm{pt}_2$, and $d(\mathrm{pt}, \mathrm{line})$ is the distance between the point and the line.
   In the case depicted in Figure 5.1 the computation in this step would yield the occlusion time $t_2$.
   (c) Compute the gradients of the intersection time according to Equation 5.20, where $T_{\nparallel}(C) = T_2$ for every $T_2 \in T_{\nparallel}(C)$ and $T_{||}(C)$ being its corresponding tangency time.
   In the case depicted in Figure 5.1 the computation in this step would consist of computing $\frac{\partial T_{\nparallel}}{\partial C}(C)$ where $T_{\nparallel}(C) = t_2$ and $T_{||}(C) = t_1$.

2. We now construct our continuous variant of the viewability matrix. Consider the sorted set $T$ of all occlusion times.

   In the case depicted in Figure 5.1 we would have
   $$T = \langle t_0, t_1, t_2, t_3 \rangle \ , \tag{5.23}$$
   We construct a table $V$ whose $i$'th entry is 1 if and only if the camera placement $C$ views (without occlusions) $x(t)$ for every $t \in [T_i, T_{i+1}]$.

   In the case depicted in Figure 5.1, where the marked camera placement views $x(t)$ in the intervals $[t_0, t_1]$ and $[t_2, t_3]$ and does not view $x(t)$ in the intervals $[t_1, t_2]$ and $[t_3, t_0]$, we would have
   $$V = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{5.24}$$

3. We compute the gradient the objective according to equation 5.10.

We recompute the gradient as many times as necessary according to the non-linear optimizer until convergence to a local optimum.

## 5.2 The Algorithm

We now proceed to describe the algorithm in the general case of an arbitrary local visibility objective, arbitrary scene and multiple cameras. The general case is only slightly more complicated than the case described in the example, requiring two modification:

- To handle general local visibility objectives we need to compute the gradient of $T_{||}(C)$.
- We handle multiple cameras by augmenting the number of columns of the viewability matrix $V$.

We proceed by computing the expression for the gradient of the tangency time, following the same lines as in the computation of $T_{\nparallel}(C)$. We write an implicit formulation of $T_{||}(C)$ and take its derivative:

$$(x(T_{||}(C)) - C)^{\mathsf{T}} \cdot n(T_{||}(C)) = 0 \tag{5.25}$$

because of the tangency property. Taking implicit derivatives of both sides we have

$$\begin{aligned} &\left(\dot{x}(T_{||}(C)) \cdot \frac{\partial T_{||}}{\partial C}(C) - I\right)^{\mathsf{T}} \cdot n(T_{||}(C)) + \\ + \quad &\left(\dot{n}(T_{||}(C)) \cdot \frac{\partial T_{||}}{\partial C}(C)\right)^{\mathsf{T}} \cdot \big(x(T_{||}(C)) - C\big) = 0 \end{aligned} \tag{5.26}$$

and by rearranging we have

$$\begin{aligned} &\left(\frac{\partial T_{||}}{\partial C}(C)\right)^{\mathsf{T}} \cdot \underbrace{\dot{x}(T_{||}(C))^{\mathsf{T}} \cdot n(T_{||}(C))}_{=0} + \\ + \quad &\left(\frac{\partial T_{||}}{\partial C}(C)\right)^{\mathsf{T}} \cdot \dot{n}(T_{||}(C))^{\mathsf{T}} \cdot \big(x(T_{||}(C)) - C\big) = n(T_{||}(C)) \ . \end{aligned} \tag{5.27}$$

Therefore

$$\frac{\partial T_{||}}{\partial C}(C) = \frac{1}{\dot{n}(T_{||}(C))^{\mathsf{T}} \cdot \big(x(T_{||}(C)) - C\big)} \cdot n(T_{||}(C))^{\mathsf{T}} \ . \tag{5.28}$$

The raw computation of the gradient is done, though we wish to make a couple of technical substitutions that would make the expression for the gradient more interpretable. Specifically we make the substitutions

$$n(t) \mapsto \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \dot{x}(t) \tag{5.29}$$

$$\dot{n}(t) \mapsto \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \ddot{x}(t) \tag{5.30}$$

$$x(T_{||}(C)) - C \mapsto \frac{\dot{x}(t)}{||\dot{x}(t)||} \cdot ||x(T_{||}(C)) - C|| \tag{5.31}$$

and write

$$\frac{\partial T_{||}}{\partial C}(C) = \frac{||\dot{x}(t)||}{||x(T_{||}(C)) - C||} \cdot \frac{1}{\ddot{x}(T_{||}(C)) \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \dot{x}(T_{||}(C))} \cdot \dot{x}(T_{||}(C))^{\intercal} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} . \tag{5.32}$$

We now interpret the expression for the gradient of $T_{||}$ by making three observation about how we would expect $T_{||}$ to change as we slightly move $C$ in various directions.

- We expect no change in $T_{||}(C)$ as we move $C$ in the direction of $\dot{x}(T_{||}(C))$.
- The distant $C$ is from $x(T_{||}(C))$ the *smaller* should be the change in $T_{||}(C)$.
- The more aligned $\dot{x}(T_{||}(C))$ and $\ddot{x}(T_{||}(C))$, the *larger* should be the change in $T_{||}(C)$.

These observations account for the terms in equation 5.32.

We now describe the general case Optimal Camera Placement algorithm. Still, the algorithm calls an off-the-shelf non-linear optimization tool for solving the optimization problem 3.16. The crux of the algorithm is in the computation of the objective and its gradient, which we now describe in detail.

Given a scene $S$, a local visibility objective $q$ and a camera placement $\vec{C} = \langle C_1, \ldots, C_k \rangle$, we evaluate the gradient $\partial Q(\vec{C})$ (or the objective $Q(\vec{C})$) as follows:

1. For every $C_j$ we compute $T_{||}(C_j)$, $T_{\nparallel}(C_j)$ and their gradients:
   (a) We compute $T_{||}(C_j)$ by finding the roots of the functions

   $$f^i_{T_{||}}(t) = n_i(t)^{\intercal} \cdot (x_i(t) - C) \tag{5.33}$$

   (b) We then compute $T_{\nparallel}(C_j)$ using a "ray-tracing" procedure. More specifically, for every tangency time $T_1 \in T_{||}(C_j)$ (computed in the previous step) we find the zeros of the functions

   $$f^i_{T_{\nparallel}}(t) = d(x_i(t), l(C_j, x(T_1))) \tag{5.34}$$

   where $x(T_1)$ is the occlusion boundary corresponding to the occlusion time $T_1$ and $l(\text{pt}_1, \text{pt}_2)$ and $d(\text{pt}, \text{line})$ are defined as before.

(c) Compute the gradients of the tangency time according to Equation 5.32, where $T_{||}(C) = T_1$ for every $T_1 \in T_{||}(C)$.

(d) Compute the gradients of the intersection time according to Equation 5.20, where $T_{\nparallel}(C) = T_2$ for every $T_2 \in T_{\nparallel}(C)$ and $T_{||}(C)$ being its corresponding tangency time.

2. We now construct our continuous variant of the viewability matrix. Consider the sorted set

$$T = \cup_j T_{\nparallel}(C_j) \bigcup \cup_j T_{||}(C_j) = \langle t_1, t_2, \ldots, t_r \rangle \, . \tag{5.35}$$

   We construct a table $V$ whose $i, j$'th entry is 1 if and only if the camera placement $C_j$ views (without occlusions) $x(t)$ for every $t \in [T_i, T_{i+1}]$.

3. We compute the gradient the objective according to equation 5.9.

## 5.3 Implementation and Limitations

In this section we relate to implementation of our algorithm and to its limitations. Specifically, we discuss integrals evaluation, roots finding and the domain of the gradient.

We evaluate specific integrals using a fixed step size approximation. This technique is guaranteed to converge to the true value of the specific integral. We note that this may seem to resemble the finite-sum formulation of the global visibility objective. However, the step size in our case only appears in the integral evaluation and therefore it only adds linear cost to the price of solving the Optimal Camera Placement problem (that is, linear in the number of gradient and objective evaluations). In contrast, in the finite-sum approach the step size is part of the problem formulation and it adds a polynomial price of solving the Optimal Camera Placement problem (where the exponent is the problem size, i.e., the sensor model times the number of cameras).

Finding the zeros of a function appears in the computations of the tangency times and the intersection times, and is a more complicated task. The way we currently compute the roots of a function $f$ is through fitting it to a polynomial. This can be done if we know in advance the number of roots we should find, which corresponds to knowing the number of tangency times and the number of corresponding intersection times.

Perhaps the most significant limitation of the algorithm is that also the integral form of the global visibility objective is not everywhere differentiable. Recall that we did not need to address the issue of the domains of the gradients of $T_{||}(C)$ and $T_{\nparallel}(C)$ only because of the specific form of the foreshortened local visibility objective. This issue cannot be overlooked in the case of a generic local visibility objective. We now discuss in detail the domains of $\frac{\partial T_{||}}{\partial C}$ and $\frac{\partial T_{\nparallel}}{\partial C}$. The gradients of $T_{||}$ and $T_{\nparallel}$ break if either

$$\dot{x}(T_{||}(C)) \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \dot{x}(T_{\nparallel}(C)) = 0 \ , \tag{5.36}$$

or

$$\ddot{x}(T_{||}(C)) \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \dot{x}(T_{||}(C)) = 0 \ . \tag{5.37}$$

Figure 5.2 illustrates the domains of the gradient of the global visibility objective in the scene from the example discussed earlier this chapter. It is insightful to note that these are exact characterizations of the conditions under which the occlusion boundaries are discontinuous with respect to the camera placement. More specifically, both conditions refer to cases where the line of sight is tangent to the object (or objects) in multiple (possibly an interval of) points. A natural way around this issue would be to solve the optimization problem within each one of the domains of the gradient. Unfortunately, the computation of the domain boundaries that correspond to satisfying Equation 5.36 require finding the roots of a two variable function. In practice we did not solve this issue, but instead we added to the computation of the gradient a validation clause to guarantee the validity of the computation. In general this approach may not be stable, however our simulations seem to suggest that the optimization still succeeds, perhaps due to the nature of the objectives.

## 5.4 Simulations

We ran a few simulations of our algorithm on the setting from the example described earlier this chapter. In the simulations we used a python implementation of the SLSQP algorithm. We compare the performance of our gradient based approach to a standard combinatorial greedy optimization approach. The greedy optimization approach consists of discretizing the camera placement space, breaking it down to 128 points on each edge of the rectangle. Then, the optimal placement of a single camera is chosen by evaluating the visibility objective at each one of the possible camera placements and picking the one with the optimal value. The optimal placement for two cameras is chosen by fixing one camera to the previously found optimal placement for a single camera and evaluating again the visibility objective for every possible placement of the second camera. The summary of the simulation results is presented in Table 5.1.

The results presented in the rows corresponding to the gradient based approach were obtained by choosing arbitrary initialization near the corners of the camera placement constraint rectangle.

The results presented in Table 5.1 clearly show that the gradient based approach finds competitive solutions to the optimization problem while requiring way less computational resources.

It is interesting to examine the measurements collected during the run of the greedy optimization algorithm. Figures 5.3 illustrates the optimal single camera placement according to the greedy algorithm, and Figures 5.4 illustrates the optimal two cameras placement according to the greedy algorithm. Figure 5.5 illustrates all the evaluations of the visibility objective made throughout the run of the greedy algorithm, where the lower curve corresponds to the measurements of the singe camera objective and the upper curve corresponds

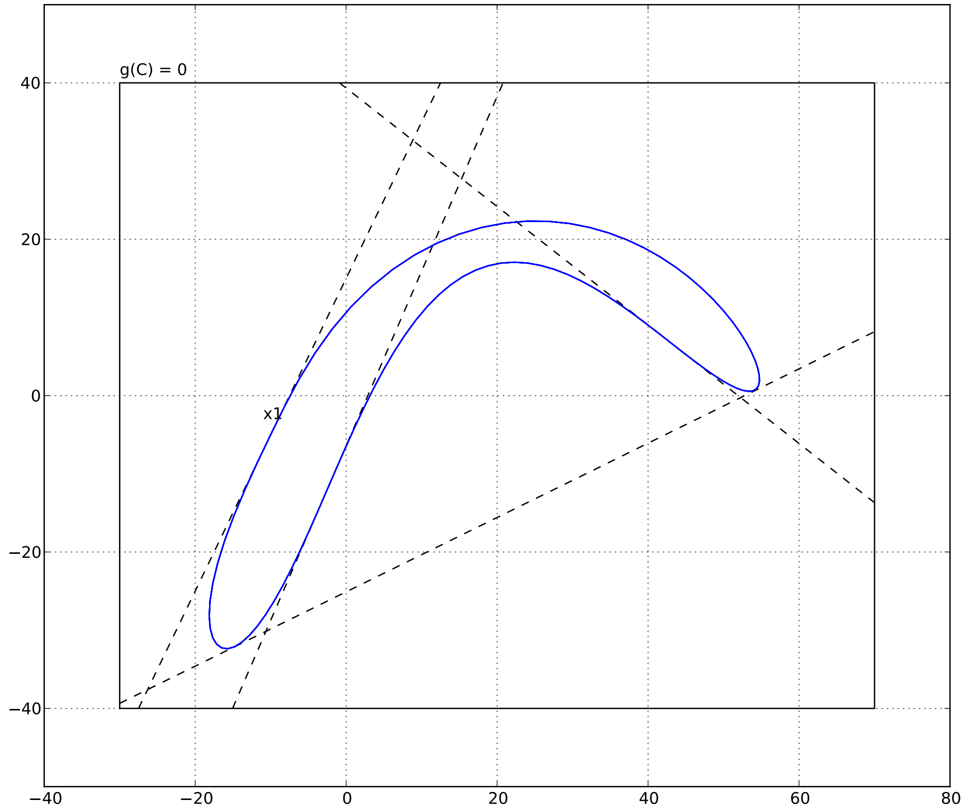

Figure 5.2. The domains on which the gradient of the occluding boundaries exist.

to the measurements of the two-cameras objective. It is interesting to notice the sharp kink in the curve in Figure 5.5, which corresponds to the camera placement illustrated in Figure 5.6.

The plot clearly shows the non-convexity of the objective, which may result with convergence to a sub-optimal solution. In practice, many of the initializations we tried converged to the true optimum, suggesting that finding the true optimum does not require an extremely high number of initializations.

| Algorithm | # of Cameras | # of Objective Evaluations | Result |
|---|---|---|---|
| Greedy | 1 | $128 \times 4 = 512$ | 87.45 |
| Greedy | 2 | $2 \times 128 \times 4 + 128 \times 4 = 1536$ | 128.48 |
| Gradient | 1 | 10 | 87.74 |
| Gradient | 2 | 16 | 121.87 |

Table 5.1. Comparing Gradient Based Approach to the Combinatorial Greedy Approach.

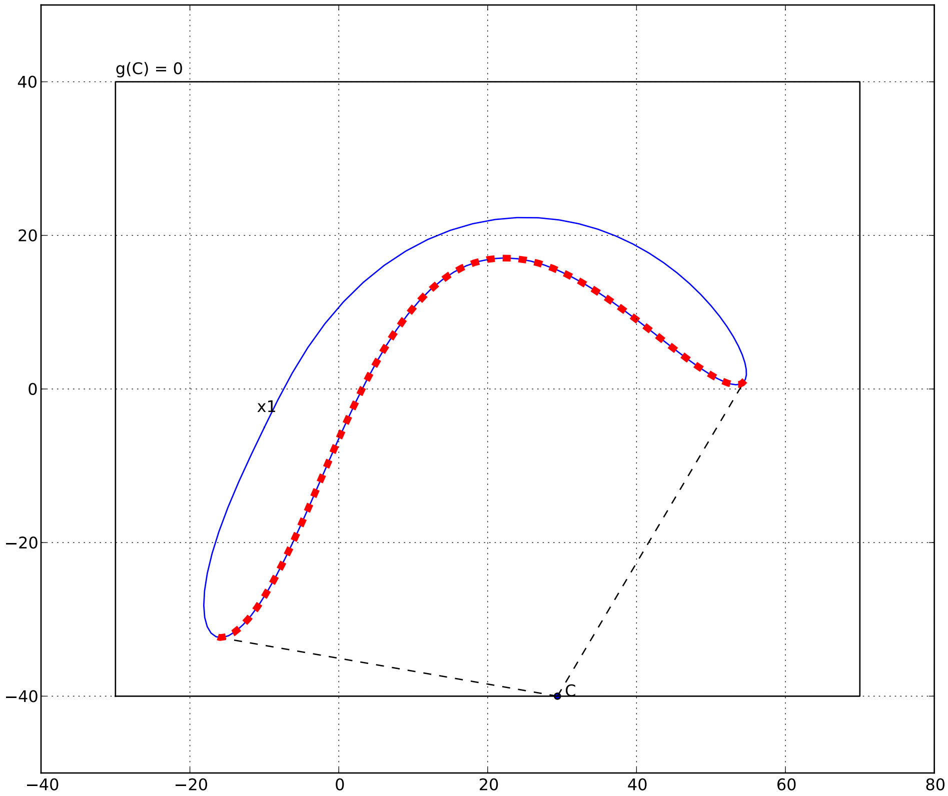

Figure 5.3. The optimal single-camera placement according to greedy algorithm.

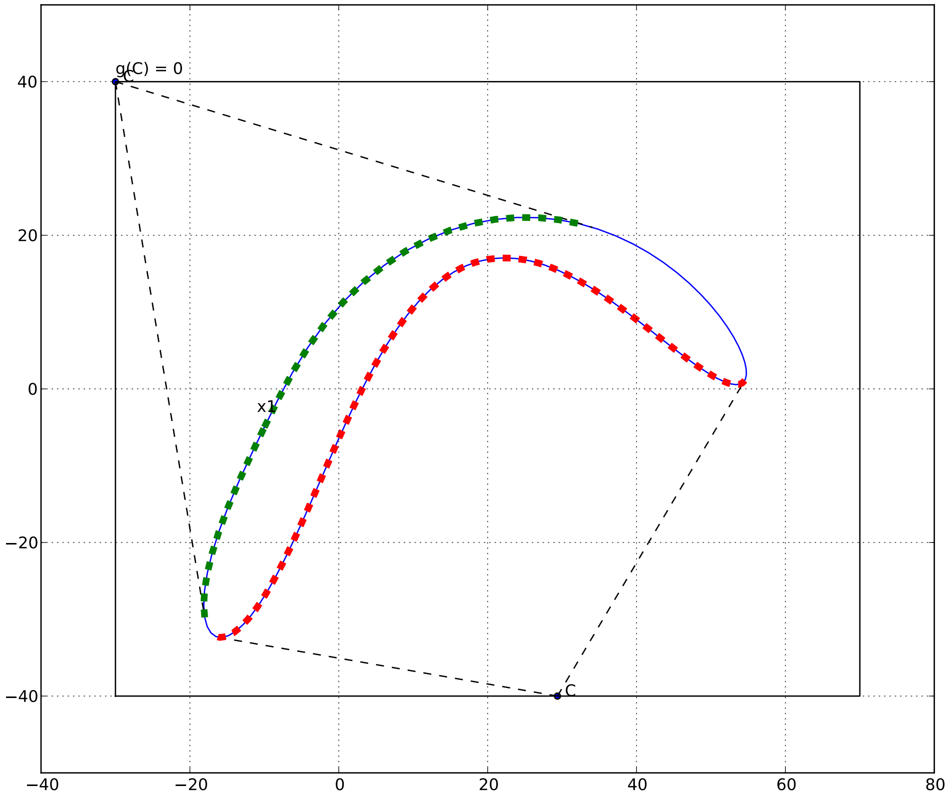

Figure 5.4. The optimal two-cameras placement according to the greedy algorithm.

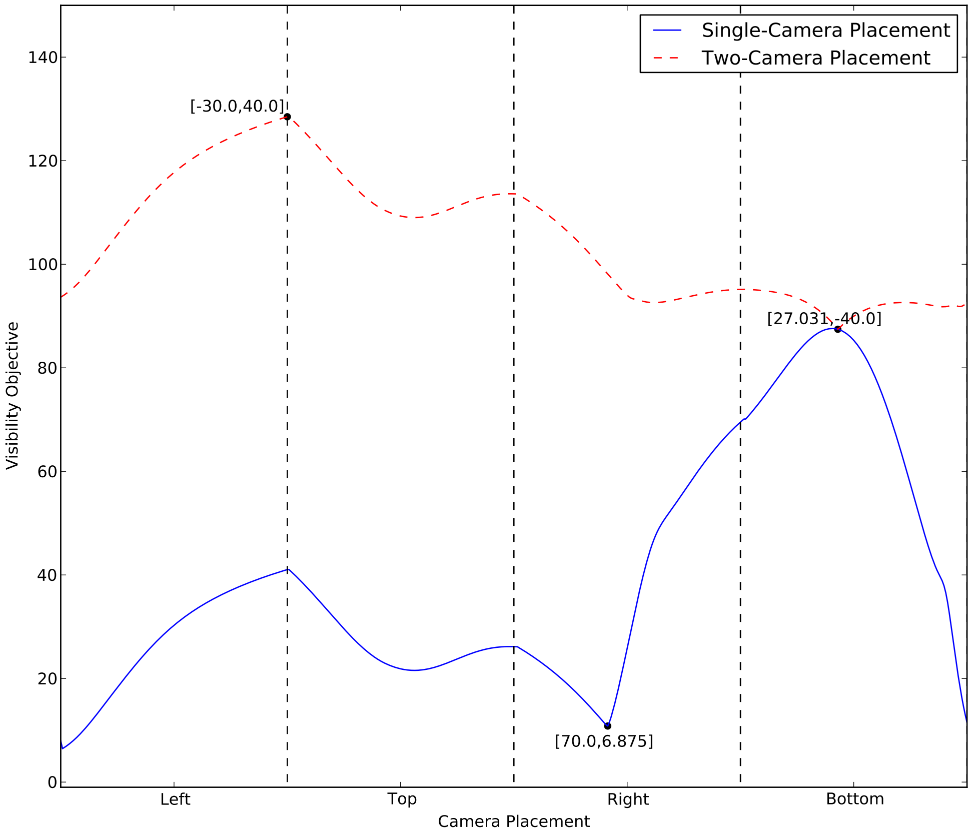

Figure 5.5. All the measurements of the objective collected throughout the run of the greedy algorithm. The lower curve corresponds to the measurements of the singe camera objective and the upper curve corresponds to the measurements of the two-cameras objective.

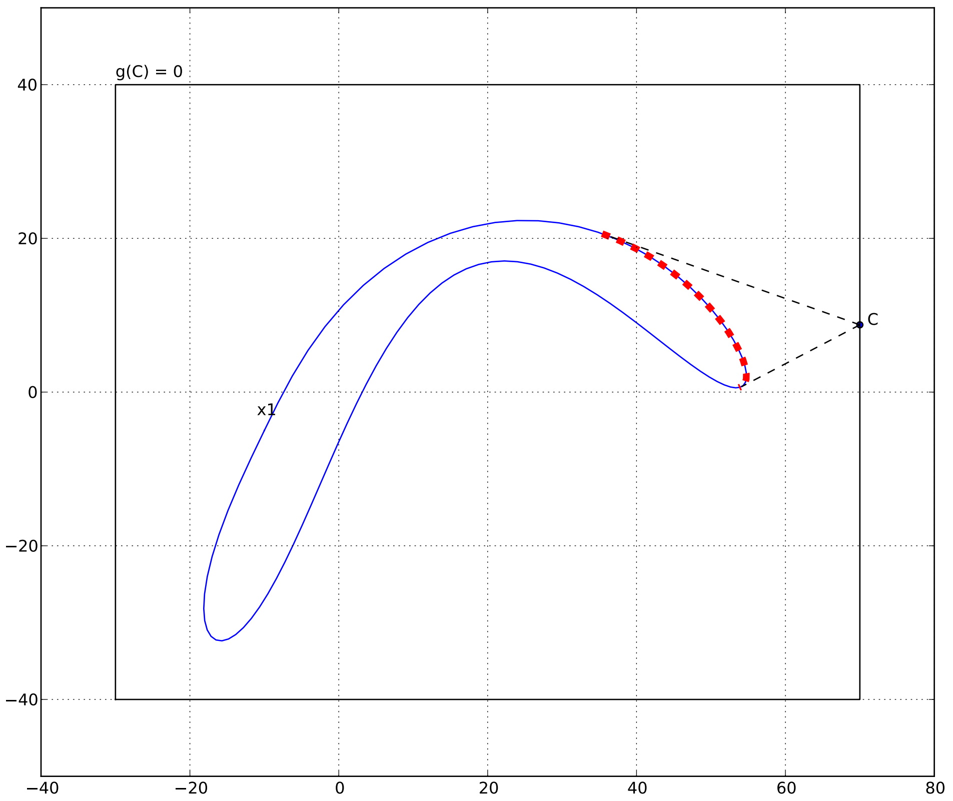

Figure 5.6. The single camera placement corresponding to the “kink” in the single-camera objective curve.

# Chapter 6

# Future Work

The work we presented here could develop in multiple promising research directions. In Chapter 1 we listed some possible extensions to our work, and in this chapter we shall elaborate on one specific direction, namely an alternative formulation for the visibility objective.

We think that the evaluation of the alternative objective we are about to propose, as well as the evaluation of its gradient, does not require the assumptions we made regarding the geometry of the target objects (in the form of knowing the number of inflection points of the object's parametrization)and that it would differentiable everywhere.

For simplicity of notation we shall discuss only the case of a single camera placement. Recall the formulation we used for the visibility indicator from Equation 3.7:

$$\chi_S(t,C) = 1 - \max_{t' \in S^1 \setminus t} \left\{ \mathbf{1}_{||x(t)-C||>||x(t')-C||} \cdot \mathbf{1}_{(x(t)-C)^\intercal \cdot (x(t')-C) = ||x(t)-C|| \cdot ||x(t')-C||} \right\} . \tag{6.1}$$

We note that the computation of $\chi_S(t)$ requires searching for the single occluding point $x(t')$. However, in any realistic sensor model occlusions do not arise from a single occluding point but from an *occluding neighborhood.* This observation gives rise to a "smoothened" formulation, where the computation of $\chi_S(t,C)$ would consist of finding a point $x(t')$ that is $\delta$-occluding in a sense that would become clear immediately. The smoothed formulation is obtained by substituting the sharp indicator functions by smooth threshold functions (for example, the sigmoid function would be applicable), which we denoted here $f$. Additionally, the maximum over the continuous interval is substituted by a threshold over an integral. More specifically, we make the substitutions (a substitution that is valid because the values the maximization objective are either 0 or 1):

$$\mathbf{1}_{x>y} \mapsto f(x-y) \tag{6.2}$$

$$\mathbf{1}_{x=y} \mapsto f(x-y+\delta) \tag{6.3}$$

$$\max_{\tau} \left\{ F(\tau) \right\} \mapsto f\left( \int_{\tau} F(\tau) \right) \tag{6.4}$$

and obtain the expression

$$\chi_S(t,C) = 1 - f\left( \int_{t' \in S^1 \setminus t} f\Big( ||x(t) - C|| - ||x(t') - C|| \Big) \cdot \right.$$
$$\left. f\Big( (x(t) - C)^\intercal \cdot (x(t') - C) - ||x(t) - C|| \cdot ||x(t') - C|| + \delta \Big) \partial t' \right) \, . \tag{6.5}$$

The expression in Equation 6.5 requires another last modification. In its current form only a single point in every object would not be considered occluded. To see this, note that every point is surrounded by a small neighborhood of points that are all $\delta$ co-aligned with it. We call these false $\delta$-occlusion relations *trivial occlusions*, as they are caused merely by the integral formulation we chose. To eliminate the trivial occlusions we restrict the limits of the integration as follows

$$\chi_S(t,C) = 1 - \max_{\epsilon_l, \epsilon_r} f\left( \int_{t' \in S^1 \setminus (t - \epsilon_l, t + \epsilon_r)} f\Big( ||x(t) - C|| - ||x(t') - C|| \Big) \cdot \right.$$
$$\left. f\Big( (x(t) - C)^\intercal \cdot (x(t') - C) - ||x(t) - C|| \cdot ||x(t') - C|| + \delta \Big) \partial t' \right) \tag{6.6}$$

subject to

$$\max_{t' \in [t - epsilon_l, t]} (x(t) - C)^\intercal \cdot (x(t') - C) - ||x(t) - C|| \cdot ||x(t') - C|| \geq -\delta \tag{6.7}$$
$$\max_{t' \in [t, t + epsilon_r]} (x(t) - C)^\intercal \cdot (x(t') - C) - ||x(t) - C|| \cdot ||x(t') - C|| \geq -\delta \, . \tag{6.8}$$

We think that this alternative objective, though less intuitive, may capture accurately the notion of neighborhood occlusion and that it would circumvent the need to compute the occlusion boundaries, thus removing the necessity of the geometric assumptions.

# Bibliography

Robert Bodor, Andrew Drenner, Paul R. Schrater, and Nikolaos Papanikolopoulos. Optimal camera placement for automated surveillance tasks. *Journal of Intelligent and Robotic Systems*, 50(3):257–295, 2007.

Samira Bouyagoub, David R. Bull, Cedric Nishan Canagarajah, and Andrew R. Nix. Automatic multi-camera placement and optimisation using ray tracing. In *ICIP*, pages 681–684. IEEE, 2010. ISBN 978-1-4244-7994-8.

Xing Chen and James Davis. An occlusion metric for selecting robust camera configurations. *Mach. Vis. Appl.*, 19(4):217–222, 2008.

C. K. Cowan. Model-based synthesis of sensor location. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 900–905, Philadelphia, PA, USA, April 1988. IEEE.

C. K. Cowan and A. Bergman. Determining the camera and light source location for a visual task. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 509–514, Scottsdale, AZ, USA, May 1989. IEEE.

C. K. Cowan and P. D. Kovesi. Automatic sensor placement from vision task requirements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):407–416, May 1988.

Cregg K. Cowan and Bharath R. Modayur. Edge-based placement of camera and light source for object recognition and location. In *ICRA (2)*, pages 586–591, 1993.

Stephan Eidenbenz, Christoph Stamm, and Peter Widmayer. Inapproximability of some art gallery problems. In *CCCG*, 1998.

Shachar Fleishman, Daniel Cohen-Or, and Dani Lischinski. Automatic camera placement for image-based modeling. *Comput. Graph. Forum*, 19(2):101–110, 2000.

Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, pages 1650–1654. AAAI Press, 2007. ISBN 978-1-57735-323-2.

Jay Lee. Analyses of visibility sites on topographic surfaces. *International Journal of Geographical Information Science*, 5(4):413–429, 1991.

Jasna Maver and Ruzena Bajcsy. Occlusions as a guide for planning the next view. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(5):417–433, 1993.

Andreas Nüchter, Hartmut Surmann, and Joachim Hertzberg. Planning robot motion for 3d digitalization of indoor environments. In *In Proc. of the 11th International Conference on Advanced Robotics (ICAR*, pages 222–227, 2003.

Gustavo Olague and Roger Mohr. Optimal camera placement for accurate reconstruction. *Pattern Recognition*, 35(4):927–944, 2002.

Joseph O'Rourke. *Art gallery theorems and algorithms*. Oxford University Press, Inc., New York, NY, USA, 1987. ISBN 0-19-503965-3.

Richard Pito. A solution to the next best view problem for automated surface acquisition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(10):1016–1030, 1999.

M. Schwager, B. Julian, M. Angermann, and D. Rus. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561, September 2011a.

M. Schwager, D. Rus, and J. J. Slotine. Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment. *International Journal of Robotics Research*, 30(3): 371–383, March 2011b. doi: http://dx.doi.org/10.1177/0278364910383444.

K. Tarabanis and R. Y. Tsai. Computing viewpoints that satisfy optical constraints. In *Proceedings of the IEEE Computer Society Conference onComputer Vision and Pattern Recognition*, pages 152–158, Maui, HI, USA, June 1991.

K. Tarabanis, R. Y. Tsai, and P. K. Allen. Automated sensor planning for robotic vision tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 76–82, Sacramento, CA, USA, April 1991.

K.A. Tarabanis, P.K. Allen, and R.Y. Tsai. A survey of sensor planning in computer vision. *Robotics and Automation, IEEE Transactions on*, 11(1):86 –104, feb 1995. ISSN 1042-296X. doi: 10.1109/70.345940.

Konstantinos A. Tarabanis, Roger Y. Tsai, and Anil Kaul. Computing occlusion-free viewpoints. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(3):279–292, 1996.

Glenn H. Tarbox and Susan N. Gottschlich. Planning for complete sensor coverage in inspection. *Computer Vision and Image Understanding*, 61(1):84–111, 1995.