

Path Planning Approach to Visual Servoing with Feature Visibility Constraints: A Convex Optimization based Solution

A.H. Abdul Hafez ^{1,2}, Anil Kumar Nelakanti ¹, and C.V. Jawahar ¹

Abstract—This paper explores the possibility of using convex optimization to address a class of problems in visual servoing. This work is motivated by the recent success of convex optimization methods in solving geometric inference problems in computer vision. We formulate the visual servoing problem with feature visibility constraints as a convex optimization of a function of the camera position *i.e.* the translation of the camera. First, the path is planned using potential field method that produces unconstrained but straight line path from the initial to the desired camera position. The problem is then converted to a constrained convex optimization problem by introducing the visibility constraints to the minimization problem. The objective of the minimization process is to find for each camera position the closest alternate position from which all features are visible. This algorithm ensures that the solution is optimal. This formulation allows the introduction of more constraints, like joint limits of the arm, into the visual servoing process. The results have been illustrated in a simulation framework.

I. INTRODUCTION

Convex optimization has been widely accepted as a powerful tool to solve many engineering problems. Its use has been extensively explored for solving a family of geometric reconstruction problems in computer vision. A wide variety of computer vision problems can be reformulated as convex optimization problems by some algebraic manipulations. Camera calibration [1], structure from motion [2], [3], [4], [5], and image registration [6] are examples of computer vision problems which are reformulated and solved efficiently as convex optimization. Convex functions, most importantly, do not have the pitfall of local minima assuring that the achieved solution always happens to be the optimal. They employ efficient real-time algorithms for optimization like sub-gradient methods, interior-point methods, etc. These advantages make convex optimization highly preferable when compared to the traditional methods like Newton’s method. The aim of this paper is to explore the utility of convex optimization to address a class of problems in visual servoing.

Visual Servoing is the problem of positioning the robotic arm or a camera relative to a set of target features. As a minimization problem, visual servoing is to move the robot end-effector from an initial pose $\mathcal{P} \in R^3 \times SO(3)$ to reach a desired pose \mathcal{P}^* . In other words, the problem is to minimize an error vector $e(s)$ of visual features $s(\mathcal{P})$ by finding a

velocity vector V that minimizes a cost function. Regulating the task function to zero indicates the task completion. Constraints may be added to the minimizer domain to satisfy requirements like keeping features visible during the servoing process. This work formulates and analyzes the problem of visual servoing in the context of convex optimization to see how these methods can help in solving the problem. Convex optimization can produce a control signal with global stability. In addition, it results a global path planning method for visual servoing. Solving the visual servoing problem by either of these two methods using convex optimization requires a convex function of the rotation and translation parameters. Convex functions of the translation can be directly constructed while convex function of rotation is still not yet derived. We consider feature visibility constraints to find a path that is closest to the straight line path by minimizing a function of only the camera position *i.e.* the translation vector of the camera pose, while keeping the rotation a constant as computed by the geodesic planner.

In this paper, we use convex optimization to plan a path that ensures the globally optimal solution while all the features used are always visible in the image. Path planning includes the computation of the intermediate poses between the initial and desired poses. In addition, the method ensures that the camera path in the Cartesian space is the closest to the straight line path with visibility constraints being satisfied which in fact is the shortest possible camera path with all image features being visible. The main advantage is the global stability of the path in addition to the flexibility of introducing different kinds of constraints like feature visibility and mechanical constraints to the servoing process by simply adding a new set of constraints to the optimization formulation.

Path planning methods [7], [8], [9] for visual servoing are based on the principle that takes advantage of the local properties like stability and robustness of the basic methods, mainly the Image Based Visual Servoing (IBVS) method. A proper image trajectory is planned in the image space to be followed by IBVS. The desired image is varying with time along the planned path to keep a small displacement between the current and desired features. In fact, there are many works in the literature that take care of the visibility of a single image point [10], [9], [11]. If the camera is within a small distance from the object, most of the features will go out of the camera field of view. However, the work presented in [12] keeps all features visible in the camera field of view during the servoing process without any knowledge about the camera path.

¹ Center for Visual Information technology, International Institute of Information Technology, Gachibowli, Hyderabad-500032, India {anil_n@students., jawahar@}iiit.ac.in

² Dept. of Computer Science and Engineering, University college of Engineering, Osmania University, Hyderabad-500007, India hafezsyar@ieee.org

Here, the visibility constraints are introduced as a linear constraints to convex optimization problem. The objective of the optimization problem is to find for each camera position the nearest alternative one with minimum distance to the original one that satisfies the visibility constraints. The global stability is ensured by the effect of two factors. First, the unconstrained path planning based on only attractive potential field is a global path planning method. Second, the optimization process ensures that the involved constraints are satisfied by the alternative position. Results and performance of the planner in image and Cartesian spaces are illustrated using a simulation framework.

II. PRELIMINARIES AND BACKGROUND

We briefly introduce the fundamentals related to the imaging process. More elaborate discussion is available at [13], [14]. Then we summarize the background material related to convex optimization. Details can be seen in [15].

A. Imaging Process

The imaging process using the perspective (Pin-hole) model can be described as follows. Assume we have the 3D point M given by the non-homogeneous coordinates in the camera frame

$$M = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \quad (1)$$

The perspective projection \hat{m} of the 3D point M onto the normalized image plane can be written as

$$\left[x = \frac{X}{Z}, y = \frac{Y}{Z} \right] \iff \hat{m} = \frac{1}{Z} \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} M \\ 1 \end{pmatrix}. \quad (2)$$

Here, the vector $\hat{m} = (x, y, 1)^T$ gives the homogeneous coordinates of the projection of the point M onto the normalized image plane.

The normalized image coordinates (expressed in meter) can be transformed to the physical image coordinates $m = (u, v, 1)^T$ (expressed in pixels) using a suitable transformation. This transformation can be found by

$$m = \mathcal{K}\hat{m}, \quad (3)$$

where

$$m = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \text{ and } \mathcal{K} = \begin{pmatrix} \alpha & -\alpha \cot \theta & u_0 \\ 0 & \frac{\beta}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (4)$$

The set of parameters $\{\alpha, \beta, u_0, v_0, \theta\}$ are the camera *intrinsic parameters*.

B. Convex Optimization Background

A set is convex if the line segment between every two pair of points in the set is contained by the set [15]. Mathematically, if C is a *convex set* and $x, y \in C$, then $\theta x + (1 - \theta)y \in C$ for $0 \leq \theta \leq 1$. A *convex hull* of a set is the smallest convex set that contains the set and is composed of all the convex combinations of the elements of the set. A set C is a cone if $\forall x \in C$ and $\theta \geq 0$, $\theta x \in C$. If the set also

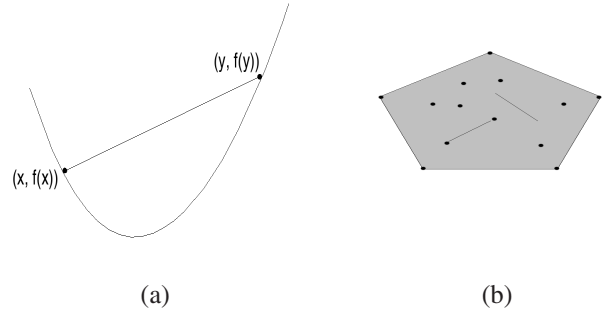


Fig. 1. The convexity of a function over a convex set in (a) and the convex hull of a set in (b).

happens to be convex then it will be a *convex cone* satisfying $\theta_1 x + \theta_2 y \in C \forall x, y \in C$ and $\theta_1, \theta_2 \geq 0$.

A function $f : R^n \rightarrow R$ is a *convex function* if its domain, $dom.f$ is convex and $\forall x, y \in dom.f$ and $0 \leq \theta \leq 1$

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y) \quad (5)$$

The above equation geometrically means that the region bounded by the line segment between $(x, f(x))$ and $(y, f(y))$ and the curve f lies above the curve. The functions include all linear functions and a few non-linear functions.

An optimization problem is convex if it is of the form

$$\min_x f_0(x) \quad (6)$$

s.t. ,

$$f_i(x) \leq b_i, \forall i = 1, 2, \dots, m \quad (7)$$

where $x \in R^n$,

$f_i : R^n \rightarrow R$ are all convex functions

and $b_i \in R \forall i = 0, 1, \dots, m$

Convex functions most importantly do not have the pitfall of local minima assuring that the achieved solution always happens to be optimal. They have efficient algorithms for optimization like sub-gradient methods, interior-point methods and others. A wide variety of problems can be rewritten as convex optimization problems by simple manipulations.

For example, Linear Programming (LP) problems are the simplest form of convex optimization problems for which both the objective function as well as the constraints are linear. In this case the objective function being linear is convex and so are the constraints and hence if a minima is achieved it has to be the global minima. There can exist at most one local minima except in the case of a constant functions where every point is a minima.

A Second Order Cone Programming (SOCP) problem is also a convex optimization problem with a linear objective function minimized over a set of affine and quadratic constraints.

$$\min_x f^T x$$

s.t. ,

$$\|A_i x - b_i\| \leq c_i^T x + d_i, \text{ for } i = 1, 2, \dots, m$$

where, $x, c_i \in R^n, A_i \in R^{n_i \times n}$,

$b_i \in R^{n_i}$ and $d_i \in R$ and $f \in R^n$.

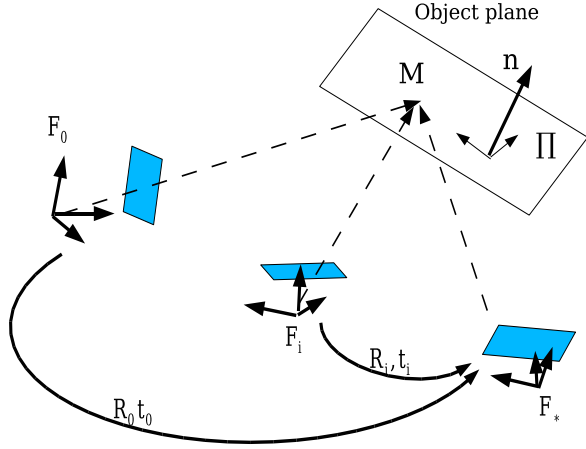


Fig. 2. 3D configuration of the planning process.

LP is a special case of SOCP. A detailed study of SOCP problems is available at [16].

Quadratic Programming (QP) problems are the problems where the objective functions are of the form $f(x) = \frac{1}{2}x'Ax + b'x$ with inequality and equality constraints as $Cx \preceq d$ and $Ex = f$. The case where A is positive semidefinite makes the objective function convex. If the constraints also happen to be quadratic along with the objective function but all convex then the problem is called (convex) Quadratically Constrained Quadratic Programming (convex QCQP) problem. These problems can be reformulated to be solved as SOCP problems.

III. PATH PLANNING FORMULATION FOR VISUAL SERVOING

Visual servoing can be solved as path planning problem. Our goal is to plan a path closest to the straight line path from a given initial position to the destination position while satisfying certain constraints like keeping the features visible all along the traversal.

Let us have an object formed by a set of N 3D points, denoted by M_o , defined in the object frame F_o . The camera frame in the initial, current, and desired frames is denoted by F_0 , F_i , F_* respectively. The transformation *T_i is defined by the rotation *R_i and translation *t_i between the current camera frame F_i and the desired frame F_* . The transformation *T_i is represented by the 6×1 vector $\mathcal{P}_i = [{}^*t_i, (u\theta)_i]^T$, where u_i is the axis and θ_i is the angle of the rotation *R_i . We thus have $\mathcal{P}_* = 0_{1 \times 6}$.

Our aim is to construct a path

$$\mathcal{L} = \{\mathcal{P}_i \mid i = 1, \dots, C\}. \quad (8)$$

In fact, this path consists of C camera configurations starting from the initial one \mathcal{P}_0 and ending with the desired one \mathcal{P}_* passing through the intermediate configurations \mathcal{P}_i .

A. The Trajectory in the Image Space

The next step is to compute the trajectory in the image space. The desired image trajectory is the projections m_i

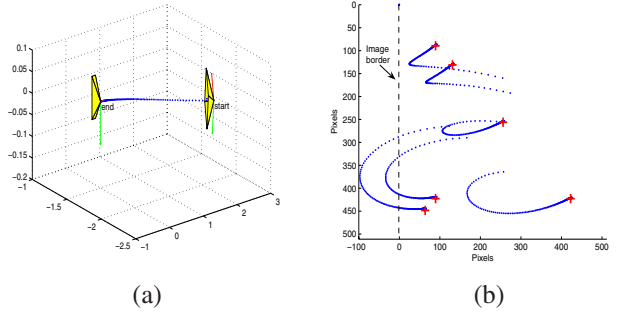


Fig. 3. The unconstrained path. The camera trajectory in (a), and the image trajectory in (b)

of the 3D points M_o in the object frame to the image. Indeed, the trajectory in the image space is defined as $\mathcal{S} = \{s_i^n \mid n = 1, \dots, N\}$. Here, s_i is the feature vector of the image point coordinates, which is given as

$$s_i = [m_i^1, \dots, m_i^N]^T.$$

The projection to the image space is done using a perspective camera model with projection matrix \mathcal{K} as follows

$$m_i^m = \mathcal{K} {}^i T_o M_o^i,$$

where the transformation ${}^i T_o$ is defined by the rotation ${}^i R_o$ and translation *t_o between the object frame F_o and the current camera one F_i .

B. Unconstrained Global Planner

If there are no constraints like collision avoidance or visibility of features to be considered the path is an unconstrained path. Unconstrained path planning problem can be efficiently solved based on the potential field method [8], [17]. The planner works by considering that the robot motion is under influence of an artificial attractive field V pulling the robot end-effector or the camera towards the target pose \mathcal{P}_* . Consider the artificial force $F(\mathcal{P})$ which is induced by the potential field V and is defined as

$$F(\mathcal{P}) = - \left(\frac{\partial V}{\partial \mathcal{P}} \right)_{\mathcal{P}_i}^T. \quad (9)$$

The incremental equation of the discrete-time planner is

$$\mathcal{P}_i = \mathcal{P}_{i-1} + \gamma_{i-1} \frac{F(\mathcal{P}_{i-1})}{\|F(\mathcal{P}_{i-1})\|}. \quad (10)$$

By selecting the artificial field as a parabolic function [8] as

$$V(\mathcal{P}_i) = \frac{1}{2} \|\mathcal{P}_i\|^2,$$

we have only one minima which is the global minima. Thus, we can write (10) as

$$\mathcal{P}_i = \mathcal{P}_{i-1} - \gamma_{i-1} \frac{\mathcal{P}_{i-1}}{\|\mathcal{P}_{i-1}\|}. \quad (11)$$

Here, γ is a constant scalar that determines the step size of the path segment.

As shown in Fig. 3.(a), the resulting path is a straight line trajectory in the Cartesian space. However, the image features may go out of the camera field of view as shown

in Fig. 3.(b). Additional processing is needed to modify the path in such a way that make all the features are visible with a minimum deviation from the straight line path.

In Section IV, we show how second order cone programming can be used to compute the new camera position which satisfies the feature visibility constraints with minimum distance to the original camera position. Thus, SOCP program is introduced in the planning process. For every intermediate path segment C_i and pose \mathcal{P}_i in which the visibility constraints are violated, the SOCP program will be activated to compute a new path segment \hat{C}_i and pose $\hat{\mathcal{P}}_i$. The new camera position satisfies the visibility constraints while keeping $\|C_i - \hat{C}_i\|$ minimum.

The object model is given in the form of 3D coordinates as M_o expressed in the object frame F_o . The given model of the object is used here to compute the relative object-camera pose. Given the initial and desired images, the object pose in the initial camera frame is 0T_o and desired one is *T_o . The relative pose between the initial and desired cameras is computed as ${}^*T_0 = {}^*T_o ({}^0T_o)^{-1} = [{}^*R_0 \quad {}^*\mathbf{t}_0]$. Since the initial pose $\mathcal{P}_0 = [{}^*\mathbf{t}_0, (u\theta)_0]^T$ and the desired one $\mathcal{P}_* = 0_{1 \times 6}$ are now available, the complete unconstrained path $\mathcal{L} = \{\mathcal{P}_i \mid i = 1, \dots, C\}$ can be computed using the incremental pose update criterion (11). Each new computed path segment that violates the visibility constraints will be modified using the SOCP program and the optimization problem presented in (12-16). We need to compute the image coordinates that are involved in the SOCP program.

The 3D coordinates of the point M_o are transformed to the initial camera frame F_0 using the transformation ${}^0T_o = [{}^0R_o \quad 0\mathbf{t}_o]$. It is not needed for the SOCP program to explicitly compute the image coordinates since the visibility constraints are developed in terms of the translation ${}^0\mathbf{t}_o$ and the 3D coordinates in frame F_0 directly. However, the image coordinates may be used in the design of the control law.

IV. FEATURE VISIBILITY BY SOCP TECHNIQUES

A. Problem Formulation

Let $\mathcal{P}_0 = [t_0, u\theta_0]^T$, be the initial pose defining the relative displacement between the initial and desired views. The visibility constraints of all image points are to be satisfied at both the initial and final positions. Say, the current pose is $\mathcal{P}_i = [{}^*\mathbf{t}_i, (u\theta)_i]^T$ at which some of the visibility constraints are not satisfied, then alternative pose is computed in such a way that all constraints are satisfied. The basic idea of the computation of the alternative pose is based on the following assumption:

Given a set of 3D points M^k and a camera pose $\mathcal{P}_i = [{}^*\mathbf{t}_i, (u\theta)_i]^T$ with respect to a reference frame F_0 and some points $M \in M^k$ are out of the camera field of view, there is alternative camera pose $\hat{\mathcal{P}}_i$ with the same rotation $(u\theta)_i$ and different translation $\hat{\mathbf{t}}$ that satisfy all visibility constraints.

As illustrated in Fig. 4, the problem of finding the alternative translation $\hat{\mathbf{t}}$ with minimum distance to the old one is

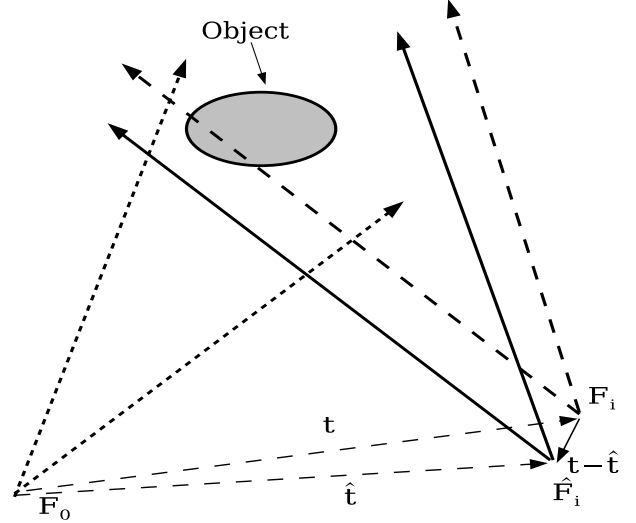


Fig. 4. Camera cones that keep the features visible.

formulated as the following minimization problem

$$\min_t \|t - \hat{t}\|^2 \quad (12)$$

s.t. ,

$$p_i(t) \geq m_0 \quad (13)$$

$$q_i(t) \geq n_0 \quad (14)$$

$$-p_i(t) + (m - m_0) \geq 0 \quad (15)$$

$$-q_i(t) + (n - n_0) \geq 0 \quad (16)$$

$$\forall i = 1, 2, \dots, N$$

where, $[p_i(t) \ q_i(t)]$ are the image points corresponding to the feature points $[X_i \ Y_i \ Z_i]$ when the translation vector of the camera position is t and the resolution of the image is $m \times n$ and the m_0, n_0 are margins from the corresponding image boundaries, see Fig. 5. This problem is a convex quadratic problem. In fact, the quadratic problem is a special case of the second order cone optimization problem. Thus, this problem can be solved efficiently using SOCP techniques.

The sketch in Fig. 4 explains the process. All the feature points can be seen from the camera pose \mathcal{P}_0 but when the camera moves to its current pose \mathcal{P}_i some of the points on the object move out of camera's field of view. Hence the translation \mathbf{t} are corrected to $\hat{\mathbf{t}}$ to get the alternative camera pose from which all the feature points are visible.

B. Convexity of the Formulation

Evidently, with a quadratic objective function to be optimized subject to a set of linear constraints, the above formulation is a Quadratic Programming (QP) problem and belongs to the class of QCQP.

Consider a camera cone, C_O , with its apex at the point O and a 3D feature point, $M \in M_o$, the feature set. The set of all points from which the point M is always visible can be defined as,

$$C'_M = \{O \mid M \in C_O\}. \quad (17)$$

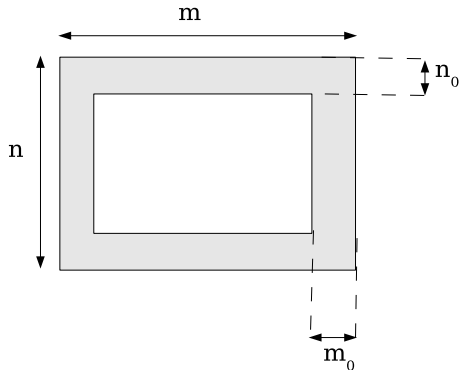


Fig. 5. Borders of the viability region in the image.

In other words, it is the set of all points that the apex O of the camera can trace with the same fixed rotation value such that the point M always belongs to the cone C_O . This set of points is nothing but the set of all possible translations of the camera for any given fixed rotation from which the point M is always visible.

Figure 6(a) shows the camera cone, C_O , at two different positions of O_1 and O_2 with the 3D point, M being visible from both the positions. Figure 6(b) shows the set C'_M corresponding to the rotational angles of camera cone C_M ($C_O, O = M$). The set C'_M can be made by reversing the directions of the rays on the boundary of the cone C_M .

The camera cone, C_O has to be a convex cone since the view frustums are always convex. Since the cone, C'_M is generated by flipping the direction of the rays of C_O about the point M , C'_M is also convex. C_O and C'_M both satisfy the definition of convex cone. The intersection of all such convex cones, C'_M for all $M \in M_0$ is the feasibility region J of the object M_0 . Since J is the intersection of convex sets and intersection is convexity preserving, the feasibility region, J is a convex set.

The objective function is the Euclidean distance between the current position and the alternative position and satisfies the inequality in (5). The constraints are all linear and also satisfy the inequality.

Now that the objective function and the constraints in the above formulation satisfy inequality (5) and their domains are convex sets, the formulation is a (convex) QCQP. This can be rewritten and solved as an SOCP as mentioned in the earlier section. Hence the formulation is an SOCP.

One may note that the formulation presented in this paper ensures that all the feature points lie inside the field of view of the camera for the alternative position computed. In fact, it is the convex hull of the points of the set M_0 that lies in the field of view. Hence, for an object to be always visible completely the convex hull of M_0 should completely enclose the object.

V. RESULTS AND EVALUATION

The proposed method for path planning algorithm using convex optimization techniques is evaluated using simulation

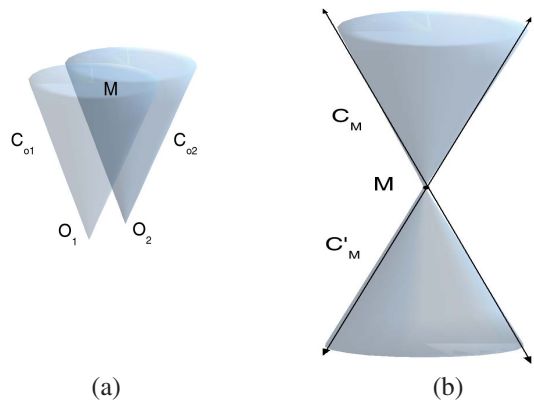


Fig. 6. The camera cones at two different points O_1 and O_2 in (a). The camera cone at 3D point M and the corresponding feasibility region for fixed rotation parameters in (b).

TABLE I
THE INITIAL AND DESIRED CAMERA POSE, AND OBJECT FRAME POSE WITH RESPECT TO THE WORLD REFERENCE FRAME.

	T_x	T_y	T_z	R_x	R_y	R_z
Initial pose	0.0	-1.5	0.0	0.0	0.0	-1.5708
Desired pose	2.1521	-2.0901	0.0	0.8	-0.6	-1.5708
Object pose	2.1521	-2.0901	0.0	0.8	-0.6	-1.5708

framework. We consider a set of points $\{M^k\}_{k=1,\dots,4}$ that belong to a non-planar object. The pixel coordinates of these points expressed in the object frame are

$$M_o = \begin{pmatrix} k & 1 & 2 & 3 & 4 \\ X & 0.25 & -0.25 & -0.25 & -0.2 \\ Y & 0.25 & 0.25 & -0.25 & -0.2 \\ Z & 0 & 0 & 0 & 0.1 \end{pmatrix}.$$

The object frame, initial and desired camera homogeneous transformations with respect to a world reference frame; and the object point coordinates are given in Tables I.

The camera parameter matrix is considered to be given as

$$\mathcal{K} = \begin{pmatrix} 1000 & 0 & 256 \\ 0 & 1000 & 256 \\ 0 & 0 & 1 \end{pmatrix}.$$

The proposed method is implemented in the simulation framework along with the unconstrained path planning method, which is based on potential field as presented in Sec. III-B. The potential field method uses only attractive field without introducing any mechanical or visibility constraints. In our method the constraints are introduced by the optimization process done using SeDuMi [18] and YALMIP [19].

The camera trajectory in the Cartesian space is illustrated in Fig 7(a) for both algorithms. The straight light camera path from the initial camera position to the desired camera position is performed by the potential field based algorithm. The camera trajectory that is performed by convex optimization based algorithm deviates from the straight line one during an intermediate part of the path in order to keep the features visible.

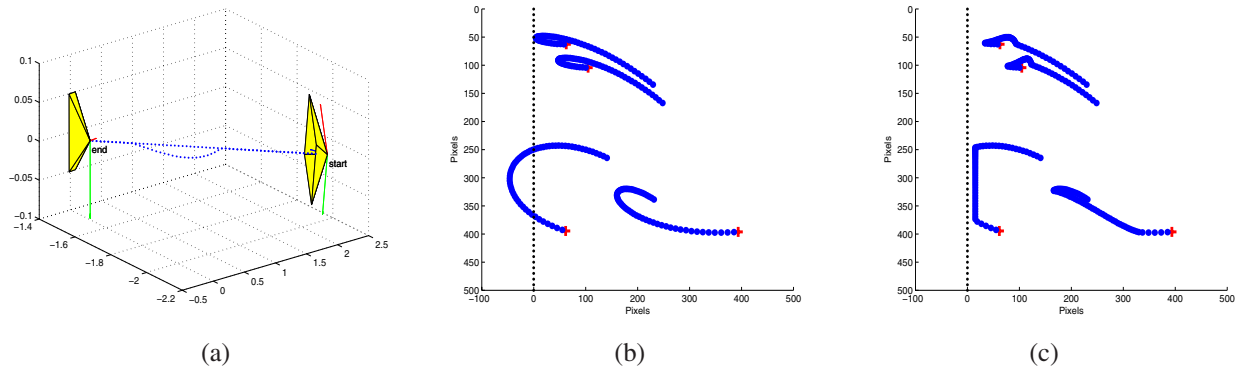


Fig. 7. The camera path in Cartesian space in (a). The straight line path is before introducing the feature visibility constraints. The image trajectory before in (b) introducing the feature visibility constraints, and after in (c).

Image trajectories corresponding to the straight line camera path are presented in Fig. 7(b). One feature has got out of the field of view. The desired positions of the features are marked by “+”. Fig. 7(c) shows the image trajectory corresponding to the camera path produced by our proposed algorithm, which is based on convex optimization. Features are successfully kept in the camera field of view. The straight line part of the image trajectory parallel to the image border corresponds to the curvature part of the camera path that appears in Fig. 7(a). The image margin is set to 20 pixels from the image border.

VI. CONCLUSIONS AND FUTURE WORKS

It has been shown in this paper that convex optimization can be applied as an efficient tool for solving problems in visual servoing. It has the advantage of providing a globally optimal solution to the problem with the set of constraints satisfied. The issue of shortest camera path along with visibility constraints has been addressed in this paper. Theoretical formulation and results of simulation justify the efficiency of this tool. Future work will focus on generalizing the cost function to include both the rotation and translation. In such case it would be possible to add the visibility and mechanical constraints to the optimization problem while providing a control signal that guides the camera to its home.

REFERENCES

- [1] B. Triggs, “Autocalibration and the absolute quadric,” in *The International Conference on Computer Vision and Pattern Recognition, CVPR’97*, June, 1997, pp. 609–614.
- [2] Q. Ke and T. Kanade, “Quasiconvex optimization for robust geometric reconstruction,” in *Proceedings of the Tenth IEEE International Conference on Computer Vision, ICCV ’05*, Beijing, China, 2005, pp. 986–993.
- [3] —, “Uncertainty models in quasiconvex optimization for geometric reconstruction,” in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR ’06*, Washington, DC, USA, 2006, pp. 1199–1205.
- [4] F. Kahl and D. Henrion, “Globally optimal estimates for geometric reconstruction problems,” in *Proceedings of the Tenth IEEE International Conference on Computer Vision, ICCV ’05*, Beijing, China, 2005, pp. 978–985.
- [5] F. Kahl, “Multiple view geometry and the L-infinity norm,” in *the Tenth IEEE International Conference on Computer Vision, ICCV’05*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 1002–1009.
- [6] M. Wachowiak and T. Peters, “High-performance medical image registration using new optimization techniques,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 2, pp. 344 – 353, Aapril 2006.
- [7] Y. Mezouar and F. Chaumette, “Model-free optimal trajectories in the image space,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS’01*, vol. 1, Maui, Hawaii, October 2001, pp. 25–31.
- [8] —, “Path planning for robust image-based control,” *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, Aug 2002.
- [9] F. Scharamm, A. Micailli, and G. Morel, “Calibration free path planning for visual servoing yielding straight line behaviour both in image and work space,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS’05*, Edmonton, Canada, August 2005.
- [10] V. Kyrki, D. Kragic, and H. I. Christensen, “New shortest-path approaches to visual servoing,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS’04*, Sendai, Japan, September-October 2004, pp. 349–354.
- [11] E. Malis, F. Chaumette, and S. Boudet, “2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement,” *International Journal of Computer Vision*, vol. 37, no. 1, pp. 79–97, June 2000.
- [12] G. Chesi, K. Hashimoto, D. Prattichizzo, and A. Vicino, “Keeping features in the field of view in eye-in-hand visual servoing: A switching approach,” *IEEE Trans. on Robotics*, vol. 20, no. 5, pp. 534–549, Oct 2004.
- [13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.
- [14] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric View point*. MIT Press, 1993.
- [15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [16] M. S. Lobo, “Applications of second-order cone programming:” [Online]. Available: citeseer.ist.psu.edu/511171.html
- [17] O. Khatib, “Real time obstacle avoidance for manipulators and mobile robots,” *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [18] J. Sturm, “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,” *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999, special issue on Interior Point Methods (CD supplement with software). [Online]. Available: citeseer.ist.psu.edu/sturm99using.html
- [19] J. Lfberg, “Yalmip : A toolbox for modeling and optimization in MATLAB,” in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: <http://control.ee.ethz.ch/~joloef/yalmip.php>