A Transformation of the Position Based Visual Servoing Problem into a Convex Optimization Problem

Yuquan Wang, Johan Thunberg and Xiaoming Hu

***Abstract*— Here we address the problem of moving a camera from an initial pose to a final pose. The trajectory between the two poses is subject to constraints on the camera motion and the visibility, where we have bounds on the allowed velocities and accelerations of the camera and require that a set of point features are visible for the camera. We assume that the pose is possible to retrieve from the observations of the point features, *i.e.*, we have a Position Based Visual Servoing Problem with constraints. We introduce a two step method that transforms the problem into a convex optimization problem with linear constraints. In the first step the rotational motion is restricted to be of a certain type. This restriction allows us to retrieve an explicit solution of the rotational motion that is optimal in terms of minimizing geodesic distance. Furthermore, this restriction guarantees that the rotational motion satisfies the constraints. Using the explicit solution, we can formulate a convex optimization problem for the translational motion, where we include constraints on workspace and visibility.**

I. Introduction

Visual servoing is used in a wide range of applications. Vision allows for non-contact measurements and brings in more information compared to other sensors such as encoders or laser sensors. Therefore, using vision in the loop, allows a robot to work in unknown and non-static environments. Traditionally visual servoing has been divided into two categories: Image Based Visual Servoing (IBVS) [2] and Position Based Visual Servoing (PBVS)[2].

A problem in visual servoing is to track a target while keeping the target inside the field of view of the camera. We refer to this problem as the visual servoing with visibility constraints. In the case of a PBVS control law the pose of the camera is measured and the camera follows a trajectory in the euclidean space. Because no control is performed in the image plane, the visibility constraints cannot be guaranteed. On the other hand, in the case of an IBVS control law, the control is performed purely in the image plane. In this case local minima may be present [4], and due to the lack of depth information, the interaction matrix has to be approximated [2].

There has been a range of different hybrid approaches to visual servoing in previous works. In these approaches PBVS and IBVS is combined. One of the classic works within this realm is the paper "$2\frac{1}{2}D$ Visual Servoing" by Malis *et.al.* [12], where homography decomposition is used. In this work a PBVS control law is used for the rotation, whereas an IBVS control law is used for the position. Even though this work is appealing it suffers from drawbacks. The homography decomposition is noise sensitive and the method does not guarantee that the visibility constraints are fulfilled. Chesi and Hashimoto [5] propose a switching controller. As long as all point features are inside the image frame a position based controller is used. Once a feature enters the boundary of the image frame, a controller consisting of either pure rotational motion or pure translational motion is used. The drawback of this switching controller is the presence of chattering. Thuilot *et.al.* [16] solves the problem by using a representation of the pose which separates the rotational and translational dynamics. Using this decoupling, the authors can solve the problem by only focusing on translational motion, where the desired trajectory in the image plane forms a straight line. However the problem of local minima is not well addressed in this approach. A method based on potential fields is proposed in [6], where a potential field is defined in the image plane. The field is used to reverse the camera motion when the features approach the boundary of the image plane.

None of the mentioned control designs gives a global solution while fulfilling the visibility constraint. Instead the methods work only locally. In addition, the problem of optimizing the camera motion is not fully addressed. In order to obtain a method that works globally, where the camera motion is optimized, trajectory planning methods should be considered.

Nonlinear optimization has been investigated for visual servoing in previous works. In [14], Toshiyuki *et.al.* propose a receding horizon approach for the stabilization of a robot arm using a position based visual controller. In each iteration a nonlinear optimization problem is solved. In [15] a nonlinear IBVS-based model predictive control is proposed. Visibility constraints, joint limitations and actuator saturations are incorporated by using the dynamic model of the robot. In [1], a nonlinear optimization problem in the image plane is formulated, where workspace limitations, visibility constraints and actuator limitations are addressed. In [8] the visual servo problem is solved as a linear quadratic (LQ) optimal control problem, either by linearizing around an equilibrium point or by feedback linearization.

The nonlinear optimization problems addressed above do not guarantee a global solution. Instead of linearizing the system around an equilibrium point or use feedback linearization, it is preferred to have a global convex problem.

In order to transform the visual servoing problem with

This work has been supported by the Swedish Foundation for Strategic Research (SSF), and the European Commission FP7 project RoboHow.Cog(FP7-ICT-288533). Yuquan Wang is with the Computer Vision and Active Perception Lab, Royal Institute of Technology, 10044 Stockholm. yuquan@kth.se. Johan Thunberg and Xiaoming Hu are with the Department of Mathematics, Royal Institute of Technology {johan.thunberg,hu}@kth.se

visibility constraints into a convex optimization problem with linear constraints, we constrain the rotational motion to be around the rotational axis between the current and the desired rotation in this paper. This restriction fulfills two purposes. Firstly the shortest rotation in terms of geodesics is around this axis, secondly an explicit solution of time is easy to obtain for the rotational motion. Using the explicit solution of the rotational motion, the translational kinematics turns into a linear time varying system. We can then easily discretize the translational kinematics and formulate it as linear equality constraints in an MPC problem.

Suppose the camera field of view is known, the visible region to the camera is obtained as a cone in $\mathbb{R}^3$. If a given set of feature points are inside this cone, the visibility constraint is fulfilled. We parameterize the visible region to each camera position along its trajectory into a set of scalar inequality constraints, which consists of the camera translation and rotation.

The proposed method can be summarized as follows. First an explicit solution is constructed for the rotational motion such that the translational kinematics is translated into equality constraints. Secondly, a convex optimization problem for the translational motion is formulated, where the visibility constraints are incorporated as inequality constraints. Compared to earlier approaches, the constraints are linear and are not only defined locally, but globally.

II. Preliminaries

In this section we introduce the mathematical framework that will be used throughout the paper. We adopt the formalism of Chaumette and Hutchinson [2], [3], and assume three different coordinate frames $\mathcal{F}_c$, $\mathcal{F}_{c^*}$ and $\mathcal{F}_0$, which correspond to the camera frame, the desired camera frame and the frame of the target respectively. The frames $\mathcal{F}_{c^*}$ and $\mathcal{F}_0$ are assumed to be static relative to each other. Let a leading superscript on a vector denote the frame it is represented in. For example, we let ${}^{c}\boldsymbol{t}_0$ and ${}^{c^*}\boldsymbol{t}_0$ be the position of the origin of the target frame in the camera frame and the desired camera frame respectively.

We use the axis-angle representation for the rotation of the camera. Let $\boldsymbol{s}$ denote the relative pose between $\mathcal{F}_c$ and $\mathcal{F}_{c^*}$. We define $\boldsymbol{s}$ as $({}^{c^*}\boldsymbol{t}_c, \theta\boldsymbol{u})$, where ${}^{c^*}\boldsymbol{t}_c$ is the position of the origin of $\mathcal{F}_c$ in the frame $\mathcal{F}_{c^*}$, and $\theta\boldsymbol{u}$ is the axis-angle representation of the rotation. We will for simplicity abbreviate ${}^{c^*}\boldsymbol{t}_c$ as $\boldsymbol{t}$ in the following. Since $\boldsymbol{s}$ denotes the relative pose, we let $\boldsymbol{s}^* = \boldsymbol{0}$ denote the desired camera pose and let $\boldsymbol{e} = \boldsymbol{s} - \boldsymbol{s}^* = \boldsymbol{s}$ denote the error between the desired camera pose and the camera pose. The axis angle representation $\theta\boldsymbol{u}$ is defined as the logarithm of the rotation matrix $\boldsymbol{R}$ and is easy to be calculated from $\boldsymbol{R}$, see *e.g* [11]. This mapping is not one-to-one on $SO(3)$, however it is one-to-one on geodesic open balls around the identity with radius less than π.

Denote the velocity by $\boldsymbol{\nu}_c = (\boldsymbol{v}_c, \boldsymbol{\omega}_c)$, where $\boldsymbol{v}_c$ is the instantaneous linear velocity of the origin of the camera frame and $\boldsymbol{\omega}_c$ is the instantaneous angular velocity of the camera frame. The dynamics of $\boldsymbol{e}$ is given by

$$\dot{\boldsymbol{e}} = \boldsymbol{L}_{\boldsymbol{e}}\boldsymbol{\nu}_c + \frac{\partial \boldsymbol{e}}{\partial t} \tag{1}$$

where $\boldsymbol{L}_{\boldsymbol{e}}$ is the transition matrix and $\frac{\partial \boldsymbol{e}}{\partial t}$ represents the target motion, *i.e.* the explicit dependence of t (that is $\frac{\partial \boldsymbol{e}}{\partial t} \neq \frac{d\boldsymbol{e}}{dt} = \dot{\boldsymbol{e}}$). In the case of a static target $\frac{\partial \boldsymbol{e}}{\partial t} = 0$. The transition matrix $\boldsymbol{L}_{\boldsymbol{e}}$ is given by

$$\boldsymbol{L}_{\boldsymbol{e}} = \begin{bmatrix} \boldsymbol{R} & 0 \\ 0 & \boldsymbol{L}_{\theta\boldsymbol{u}} \end{bmatrix}, \tag{2}$$

where $\boldsymbol{R}$ is the rotation matrix between the frame $\mathcal{F}_c$, and $\mathcal{F}_{c^*}$. The matrix $\boldsymbol{L}_{\theta\boldsymbol{u}}$ is given as

$$\boldsymbol{L}_{\theta\boldsymbol{u}} = \boldsymbol{I}_3 + \frac{\theta}{2}\hat{\boldsymbol{u}} + \left(1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2\frac{\theta}{2}}\right)\hat{\boldsymbol{u}}^2. \tag{3}$$

The function $\text{sinc}(x)$ is defined so that $x\text{sinc}(x) = sin(x)$ and $\text{sinc}(0) = 1$. The matrix $\hat{\boldsymbol{u}}$ is the skew symmetric matrix of the vector $\boldsymbol{u} \in \mathbb{R}^3$. The rotation matrix $\boldsymbol{R}$ is obtained from $\theta\boldsymbol{u}$ by Rodrigues' formula as

$$\boldsymbol{R}(\theta\boldsymbol{u}) = \boldsymbol{u}\boldsymbol{u}^T + \cos(\theta)(\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^T) + \sin(\theta)\hat{\boldsymbol{u}}. \tag{4}$$

Observe that the transition matrix (2) is block diagonal. The transition matrix is invertible for $\theta \in (-2\pi, 2\pi)$, see [12]. In [5], instead of choosing $\theta\boldsymbol{u}$ as the representation of the rotational motion, $\sin(\theta)\boldsymbol{u}$ is chosen. However, with that choice of representation, the transition matrix is only invertible for $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$.

III. Problem formulation

The problem considered in this paper is to construct a kinematic control law for a calibrated perspective eye-in-hand camera so that the error between the current pose of the camera and the desired pose of the camera, $\boldsymbol{e}$, converges to zero, *i.e.*, the camera shall move so that $\mathcal{F}_{c*}$ and $\mathcal{F}_c$ coincide, see Figure 1. This is a trajectory planning problem in $SE(3)$ with constraints. The objectives and constraints are formulated mathematically in the subsequent subsections. We introduce $T_s > 0$ as the sampling period and let the discrete time 0 correspond to the continuous time t_0.

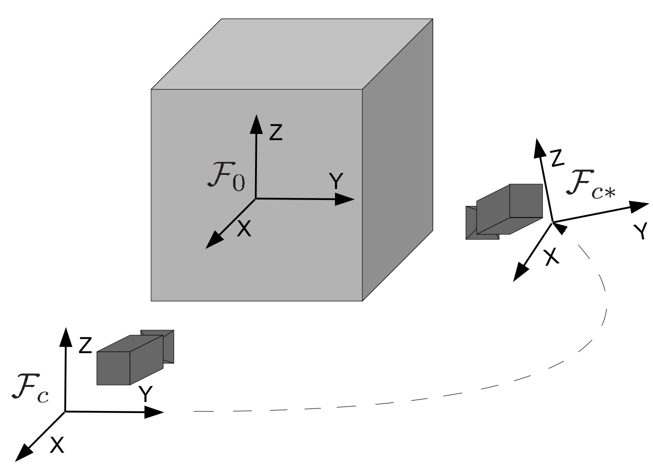

Fig. 1. The camera shall be moved from its current pose to a final pose.

A. Camera Motion Constraints

In this subsection we formulate the constraints on the velocities *i.e.*, the control signals. These constraints are formulated as bounds on the magnitudes of the control signals. Let us define $[v_1, v_2, v_3] = \boldsymbol{v}_c$ and $[\omega_1, \omega_2, \omega_3] = \boldsymbol{\omega}_c$. We consider the following bounds for the control variable $\boldsymbol{\nu}_c = (\boldsymbol{v}_c, \boldsymbol{\omega}_c)$.

$$|v_i| \leq \alpha_i, \quad |\omega_i| \leq \beta_i, \tag{5}$$

where $\alpha_i, \beta_i \in \mathbb{R}^+$, $i = 1, 2, 3$. Each bound on $\boldsymbol{v}_c$ in (5) can be written as linear constraints at time step k, *i.e.*

$$v_i(k) \leq \alpha_i \quad -v_i(k) \leq \alpha_i, \tag{6}$$

for $i = 1, 2, 3$. In order to address smoothness constraints on the translational velocity, we construct the vectors $\boldsymbol{t}_s$ and $\boldsymbol{v}_s$:

$$\boldsymbol{t}_s = [\boldsymbol{t}(0)^T, \boldsymbol{t}(1)^T, ..., \boldsymbol{t}(N-1)^T]^T$$

and

$$\boldsymbol{v}_s = [\boldsymbol{v}_c(0)^T, \boldsymbol{v}_c(1)^T, ..., \boldsymbol{v}_c(N-1)^T]^T,$$

where N is the time horizon. Let $\boldsymbol{t}^{(j)}$ denote the discrete time j:th derivative of $\boldsymbol{t}_s$. Now we can formulate magnitude constraints in a similar way as in equations (6). One can also construct a convex cost functional containing these terms as

$$f = \| \sum_{i=1}^{n} \boldsymbol{t}^{(i)} \|_2^2, \tag{7}$$

where $\boldsymbol{t}^{(i)}$ is defined as

$$\boldsymbol{t}^{(i)} = \frac{\boldsymbol{t}^{(i-1)}(t_o + (k+1)T_s) - \boldsymbol{t}^{(i-1)}(t_o + kT_s)}{T_s},$$

and $\boldsymbol{t}^{(0)} = \boldsymbol{t}$.

Remark 3.1: It is also possible to impose further smoothness constraints through solving a more general spline optimization problem, see *e.g.* [7].

B. Visibility Constraints

Here we formulate the visibility constraints and an occlusion constraint. We assume that the camera is a calibrated pinhole camera with a planar image surface and there is assumed to be n coplanar point features at the target. Each such point feature has coordinates ${}^c\boldsymbol{P}_i = [X_i, Y_i, Z_i]$, $i = 1, ..., n$, in the camera frame. Its projection onto the image plane is ${}^c\boldsymbol{p}_i = [x_i, y_i, 1]$, where x_i and y_i are defined as:

$$x_i = \frac{X_i}{Z_i}, \quad y_i = \frac{Y_i}{Z_i}. \tag{8}$$

Since the image from the camera is used as input, ${}^c\boldsymbol{p}_i$ and ${}^{c^*}\boldsymbol{p}_i$[1] are known. In order to estimate the coordinates ${}^c\boldsymbol{P}_i$ and ${}^{c^*}\boldsymbol{P}_i$ from them, we need to solve a perspective n point problem (PnP), for which there recently has appeared efficient $\mathcal{O}(n)$ computational time algorithms, see [10]. The pose $\boldsymbol{s} = (t, \theta\boldsymbol{u})$ between ${}^{c^*}\boldsymbol{P}_i$ and ${}^c\boldsymbol{P}_i$ can then be solved using the method by Horn *et.al.* [9]. Even though there exist many applications where the target model is unknown, there is a vast amount of applications, *e.g.* in mobile robotics, where a model of the target is available, justifying the assumption of a known model of the target.

The image frame is defined by the four corners ${}^c\boldsymbol{c}_1, {}^c\boldsymbol{c}_2, {}^c\boldsymbol{c}_3$ and ${}^c\boldsymbol{c}_4$, where ${}^c\boldsymbol{c}_1 = [-a, -b, 1]$, ${}^c\boldsymbol{c}_2 = [a, -b, 1]$, ${}^c\boldsymbol{c}_3 = [a, b, 1]$, ${}^c\boldsymbol{c}_4 = [-a, b, 1]$ and $a, b \in \mathbb{R}^+$. Thus, ${}^c\boldsymbol{p}_i$ is visible by the camera if

$$-a \leq x_i \leq a, \tag{9}$$
$$-b \leq y_i \leq b. \tag{10}$$

By virtue of (8), the constraints (9-10) lead to the following constraints on each ${}^c\boldsymbol{P}_i$:

$$X_i - aZ_i \leq 0, \tag{11}$$
$$-Y_i - bZ_i \leq 0, \tag{12}$$
$$-X_i - aZ_i \leq 0, \tag{13}$$
$$Y_i - bZ_i \leq 0. \tag{14}$$

Therefore each visible ${}^c\boldsymbol{P}_i$ should be contained in the convex cone defined by the origin of $\mathcal{F}_c$ and the four corners of the image frame $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3$ and $\boldsymbol{c}_4$. In Figure 2 this convex cone is illustrated. These constraints (11-14) are linear in ${}^c\boldsymbol{P}_i$ and ${}^c\boldsymbol{P}_i$ is a linear combination of $\boldsymbol{s} = (t, \theta\boldsymbol{u})$ as follows:

$${}^c\boldsymbol{P}_i = \boldsymbol{R}^T(({}^{c*}\boldsymbol{P}_i) - \boldsymbol{t}), \tag{15}$$

where $\boldsymbol{R}$ is defined in (4). Using (15), the constraints (11-14) are effectively linear in $\boldsymbol{s} = (t, \theta\boldsymbol{u})$.

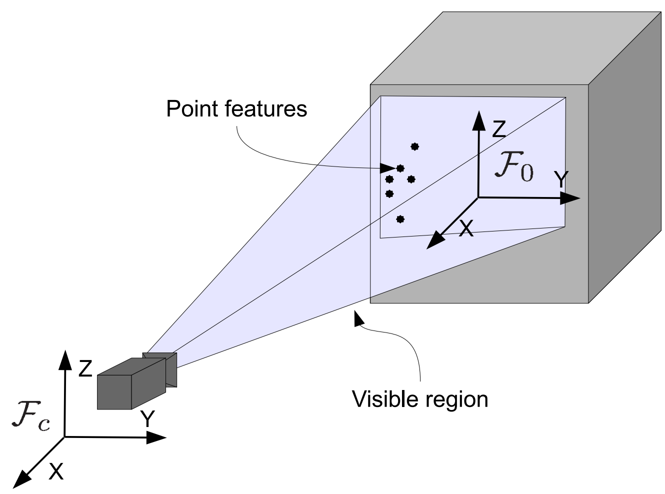

Fig. 2. An illustration of the visibility constraints in (11-14). The point features must be contained in a convex cone defined by the origin of $\mathcal{F}_c$ and the four corners of the image frame.

Unfortunately the constraints (11-14) are not sufficient ,we need to add one more occlusion constraint. The introduction of this constraint is motivated by the following situation. Assume that the coplanar point features are attached on one

[1] This is defined in the disired camera frame $\mathcal{F}_{c^*}$ as indicated by the leading superscript. Or ${}^o\boldsymbol{p}_i$, that is defined in the frame attached at the target $\mathcal{F}_0$, because the transformation between $\mathcal{F}_{c^*}$ and $\mathcal{F}_0$ is assumed to be static and known. Without this knowledge it is only possible to retrieve their positions up to scale.

side of a wall, and assume that the camera is positioned on the other side of the wall. In this situation constraints (11-14) can be satisfied, *i.e.*, all the points are in the cone, but the points cannot be seen by the camera due to the occlusion of the wall.

Let the normal to the plane pointing into the half space where the points are visible, be denoted by ${}^{c*}\boldsymbol{n}$ in the frame $\mathcal{F}_{c^*}$. An occlusion requirement in the camera frame $\mathcal{F}_c$ for each ${}^{c}\boldsymbol{P}_i$ is that

$$ {}^{c}\boldsymbol{e}_3 \boldsymbol{R}^T ({}^{c*}\boldsymbol{n}) < 0, \tag{16} $$

where ${}^{c}\boldsymbol{e}_3 = [0, 0, 1]$ and $\boldsymbol{R}^T$ represents the relative rotation between $\mathcal{F}_c$ and $\mathcal{F}_{c^*}$.

Remark 3.2: The constraint (16) is the most simple occlusion constraint we can impose. More advanced constraints can also be considered including many objects that occlude each other.

IV. Solution

The method we use to solve the problem consists of two consecutive steps. In each step a subset of the constraints (6), (11-14) and (16) are treated. After both steps a trajectory has been constructed for which all constraints are satisfied. Note that this trajectory is defined as an explicit solution of time, and in order to address the issue of measurement noise we end this section by proposing a Model Predictive Control (MPC) algorithm. The two steps in the proposed approach can be summarized as follows.

1) Construct the angular velocity $\boldsymbol{\omega}_c(t)$ while fulfilling the constraint (16). Using $\boldsymbol{\omega}_c$, an explicit solution for $\theta(t)\boldsymbol{u}(t)$ is obtained, which implies that an explicit solution for the rotation matrix $\boldsymbol{R}(t)$ can be obtained via Rodrigues' formula.
2) Using the solution for $\boldsymbol{R}(t)$ the translational dynamics is given as a time varying linear system. This linear system is then discretized such that the translation can be represented as eqaulities, expressed of its initial value and $\boldsymbol{R}(t)$. The above equalities, the linear constraints (11-14), (6) together with a convex functional form a convex optimization problem with linear constraints. This optimization problem can be solved for arbitrarily number of sampling peroids, but in the MPC-approach only the solution of the first sampling peroid is implemented in each iteration.

The two different steps will now be explained in detail in the following subsections.

A. Step 1 - Rotational Motion

The shortest geodesic distance between $\boldsymbol{R}$ and $\boldsymbol{I}$ is $|\theta| \in [0, \pi]$ in $SO(3)$, where θ is the angle in the axis-angle representation of $\boldsymbol{R}$, see [13]. This distance is also known as the Riemannian distance. The shortest path is a rotation around the axis $\boldsymbol{u}$ with magnitude $|\theta|$. Provided $\theta(t_0)\boldsymbol{u}(t_0)$ and ${}^{c*}\boldsymbol{n} = \boldsymbol{0}$ fulfills the constraint (16), it follows that (16) is fulfilled for all rotations $\theta\boldsymbol{u}$ where $\theta \in [\theta(t_0), 0]$.

Thus in order to obtain this shortest path in $SO(3)$ we can restrict the rotational motion to be around the rotational axis $\boldsymbol{u}$, *i.e.* $\boldsymbol{\omega}_c = \omega_c\boldsymbol{u}$. The design of $\boldsymbol{\omega}_c$ has thus been reduced to the problem of finding the scalar control law ω_c. When using a controller $\boldsymbol{\omega}_c$ on the form $\omega_c\boldsymbol{u}$, the dynamics in (1) become simple due to the structure of (3). All the nonlinear terms in (3) are orthogonal to $\boldsymbol{u}$ and the kinematics for the rotation is equivalent to the linear system

$$ \dot{\theta}\boldsymbol{u} = \boldsymbol{L}_{\theta\boldsymbol{u}}\boldsymbol{u}\omega_c = \omega_c\boldsymbol{u}. \tag{17} $$

A suitable way of choosing the controller ω_c is

$$ \omega_c = -k\theta, \tag{18} $$

or

$$ \omega_c = -k\text{sign}(\theta), \tag{19} $$

where $k \in \mathbb{R}^+$ is chosen so that $k|\theta(t_0)u_1| \leq \beta_1$, $k|\theta(t_0)u_2| \leq \beta_2$ and $k|\theta(t_0)u_3| \leq \beta_3$, *i.e.* the constraint (6) is satisfied. The three parameters u_1, u_2 and u_3 are the elements of $\boldsymbol{u}$. Another way of finding a suitable ω_c is by solving a linear quadratic optimal control problem with constraints $\dot{\theta} = \omega_c$.

B. Step 2 - Translational Motion

Using the solved rotational motion $\theta(t)\boldsymbol{u}$, the dynamics for the translational vector $\boldsymbol{t}$, is given by

$$ \dot{\boldsymbol{t}} = \boldsymbol{R}(t)\boldsymbol{v}_c \tag{20} $$

where $\boldsymbol{R}(t)$ is given in (4). The explicit solution for $\boldsymbol{t}(t)$ is given by

$$ \boldsymbol{t}(t) = \boldsymbol{t}(t_0) + \int_{t_0}^{t} \boldsymbol{R}(s)\boldsymbol{v}_c ds. \tag{21} $$

And its corresponding discretized dynamics is

$$ \boldsymbol{t}(k+1) = \boldsymbol{t}(k) + \boldsymbol{B}(k)\boldsymbol{v}_c(k), \tag{22} $$

where $\boldsymbol{t}(k)$ and $\boldsymbol{v}_c(k)$ is short notation for $\boldsymbol{t}(t_0 + kT_s)$ and $\boldsymbol{v}_c(t_0 + kT_s)$ at time step k. The matrix $\boldsymbol{B}(k)$ is defined as

$$ \boldsymbol{B}(k) = \int_{t_0+kT_s}^{t_0+(k+1)T_s} \boldsymbol{R}(s)ds \tag{23} $$

and $k \in \{0, 1, ...\}$.

Remark 4.1: In many cases the matrix $\boldsymbol{B}(k)$ is only defined up to quadrature, but in most of these cases the integral expression can be found with sufficient accuracy in numerical tables.

We are now ready to pose a convex optimal control problem in order to obtain the discrete translational motion $\boldsymbol{v}_c$. This problem is on the following form

$$ \left[\begin{array}{l} \min\limits_{\boldsymbol{v}_s} \ \sum\limits_{i=1}^{n} \|\boldsymbol{t}_s^{(i)}\|_2^2 + \|\boldsymbol{v}_s\|_2^2 \\ \text{ s.t. } \boldsymbol{t}(k+1) = \boldsymbol{t}(k) + \boldsymbol{B}(k)\boldsymbol{v}_c(k), \quad k = 1, 2, ..., N, \\ \qquad \boldsymbol{A}_j[\boldsymbol{t}_s^T(k), \boldsymbol{v}_s^T(k)]^T \leq \boldsymbol{b}_j, \ j = 1, 2, \ldots, M \\ \qquad \ldots \end{array} \right] \tag{24} $$

The inequalities $\boldsymbol{A}_j[\boldsymbol{t}_s^T, \boldsymbol{v}_s^T]^T \leq \boldsymbol{b}_j$ captures all constraints defined in the paper except (16).

In the precence of noise, the procedure that was described in the two preceding subsections can now be repeated at each time $t_0, t_0 + T_s, t_0 + 2T_s, ...$ until we have reached a desired convergence. The procedure is formalized into the following algorithm

Algorithm 1

1) At time t_0, solve the perspective n point problem (PnP) to get (an estimate of) the relative pose $\boldsymbol{e}(t_0) = ({}^c\boldsymbol{t}_{c^*}, \theta\boldsymbol{u})$.
2) Solve system (17) to obtain $\boldsymbol{\theta}_c(k)$ $k = 1, 2, \ldots N$. The rotation matrices $\boldsymbol{R}(k)$ could be constructed according to (4).
3) Construct state transition matrices $\boldsymbol{B}(k)$ of translational motion according to (22) and (23).
4) Solve the optimization problem (24) for time horizon N to get $\boldsymbol{v}_c(k)$. **If** no feasible solution exists, then goto 2) and solve $\boldsymbol{\theta}_c(k)$ again, **else** goto 5).
5) Use $\boldsymbol{\nu} = [\boldsymbol{\omega}_c(k)^T, \boldsymbol{v}_c(k)^T]^T$ for the time period $t \in [t_0, t_0 + NT_s]$.

V. Illustrative Example

In order to illustrate the the method we will go through one iteration of **Algorithm 1** for a specific problem. In this problem the desired pose is given by

$$\boldsymbol{s}^* = \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right).$$

Assume one set of coplaner point features are available and their coordinates ${}^{c^*}\boldsymbol{P}$ in the desired camera frame $\mathcal{F}_{c^*}$ are given. Without loss of generality, we choose the sampling period $T_s = 0.02$ second and time horizon $T = 1$ second. Then there are $N = 50$ steps to be considered.

In **step 1)** the initial pose is estimated as

$$(\boldsymbol{t}(t_0), \theta(t_0)\boldsymbol{u}(t_0)) = \left(\begin{bmatrix} 30 \\ 30 \\ 30 \end{bmatrix}, \theta_0 \begin{bmatrix} -0.6204 \\ 0.7653 \\ -0.1708 \end{bmatrix} \right),$$

where $\theta_0 = 0.9773$. In **step 2)** (17) could be solved either by (18) or (19). Then use $\boldsymbol{\theta}_c(k)$ and $\boldsymbol{u}$ to construct the rotation matrices.

In **step 3)** After $\boldsymbol{B}(k)$ in (22) is calculated using (23), $\boldsymbol{t}_s(k)$ is a linear combination of the translational velocities $\boldsymbol{v}_c(k)$.

In **step 4)** Using $\boldsymbol{t}_s(k)$, $\boldsymbol{R}(k)$ together with ${}^{c^*}\boldsymbol{P}$, coordinates of the feature points ${}^c\boldsymbol{P}$ in the camera frame $\mathcal{F}_c$ is obtained using (15). Given ${}^c\boldsymbol{P}$, the visibility constraints turns into a set of linear inequalities with respect to $\boldsymbol{v}_c(k)$.

In **step 5)** we now have all the necessary building blocks in order to formulate the optimization problem:

$$\min_{\boldsymbol{v}_s} \|\boldsymbol{v}_s\|_2^2 + \sum_{i=1}^{3} \|\boldsymbol{t}_s^{(i)}\|_2^2 \tag{25a}$$

$$\text{s.t. } \boldsymbol{t}(k+1) = \boldsymbol{t}(k) + \boldsymbol{B}(k)\boldsymbol{v}_c(k), \tag{25b}$$

$$\begin{aligned} X_i - aZ_i &\leq 0, \\ -Y_i - bZ_i &\leq 0, \\ -X_i - aZ_i &\leq 0, \\ Y_i - bZ_i &\leq 0. \end{aligned} \tag{25c}$$

$$v_i(k) \leq \alpha_i \quad -v_i(k) \leq \alpha_i, \tag{25d}$$

The simulation results are covered in the following figures:

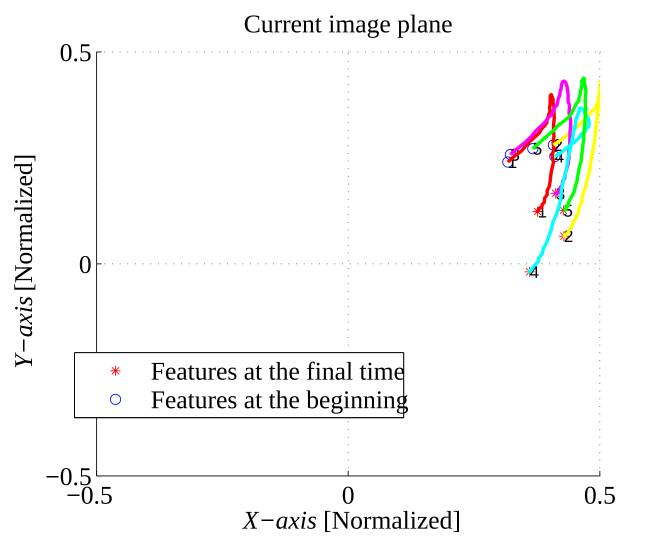

Fig. 3. Point feature trajectories on normalized image plane

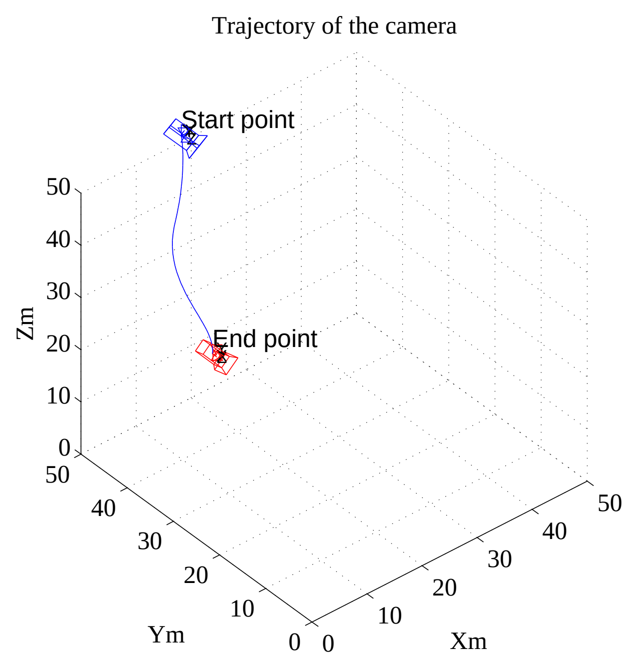

Fig. 4. Camera spatial trajectory

The point feature trajectories in the image plane are shown in Fig. 3 and the corresponding camera spatial trajectory is shown in Fig. 4. Because both the initial position and the desired position of the point features are close to the image plane boundary, it is a difficult task to keep the visibility constraint. However all the point feature trajectories stay inside the image plane.

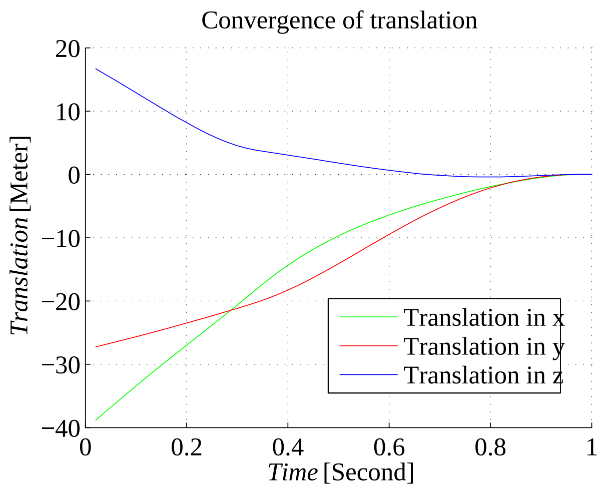

Fig. 5. Convergence of translation

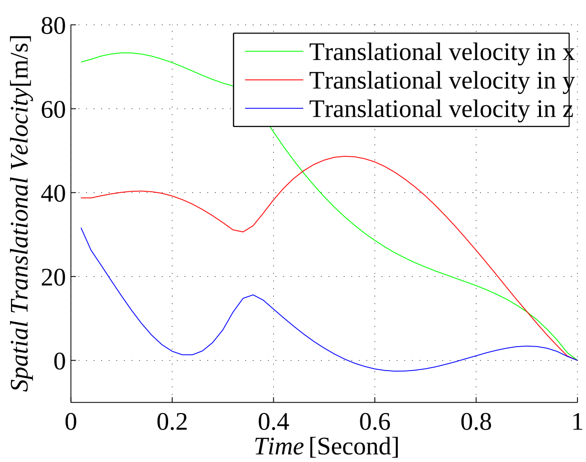

Fig. 6. Convergence of the translational velocities

The convergence of translation is shown in Fig. 5 and the corresponding translational velocity convergence is shown in Fig. 6.

VI. Conclusions and Future Work

In this paper we have proposed a new path planning method for constrained Position Based Visual Servoing (PBVS). We assume a known target with known point features. The $3D$ coordinates of the point features are obtained using the ePnP method, see [10], which has $\mathcal{O}(n)$ computational time in the number of point features n. Then the pose is obtained using the method by Horn [9].

The problem is solved in two consecutive steps, the first step is to find a feasible rotational motion and the next step is to find a feasible trajectory for the translational motion. In the second step, a convex optimization problem is formulated, where various linear constraints on the vision and the motion are incorporated.

For future work, the moving target case as well as additional constraints such as resolution constraints on the camera should be further explored.

References

[1] G. Allibert, E. Courtial, and F. Chaumette. Predictive control for constrained image-based visual servoing. *IEEE Transactions on Robotics*, 26(5):933–939, 2010.

[2] F. Chaumette and S. Hutchinson. Visual servo control. I. Basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006.

[3] F. Chaumette and S. Hutchinson. Visual servo control. II. Advanced approaches [Tutorial]. *IEEE Robotics & Automation Magazine*, 14(1):109–118, 2007.

[4] Franois Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing, 1998.

[5] 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 Transactions on Robotics*, 20(5):908–914, 2004.

[6] P.I. Corke and S.A. Hutchinson. A new partitioned approach to image-based visual servo control. *Robotics and Automation, IEEE Transactions on*, 17(4):507 –515, aug 2001.

[7] B. Demeulenaere, J. De Caigny, J. Swevers, and J. De Schutter. Dynamically Compensated and Robust Motion System Inputs Based on Splines: A Linear Programming Approach. In *American Control Conference, 2007. ACC'07*, pages 5011–5018, 2007.

[8] K. Hashimoto and H. Kimura. LQ optimal and nonlinear approaches to visual servoing. *Visual Serving: Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*, page 165, 1993.

[9] B.K.P. Horn, H.M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, 1988.

[10] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

[11] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An invitation to 3-D vision*. Springer, 2004.

[12] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 D visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, 1999.

[13] M. Moakher. Means and averaging in the group of rotations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):1–16, 2003.

[14] T. Murao, H. Kawai, and M. Fujita. Predictive visual feedback control with eye-in/to-hand configuration via stabilizing receding horizon approach. In *Proc. of the 17th IFAC World Congress on Automatic Control*, pages 5341–5346, 2008.

[15] M. Sauvee, P. Poignet, E. Dombre, and E. Courtial. Image based visual servoing through nonlinear model predictive control. In *Decision and Control, 2006 45th IEEE Conference on*, pages 1776–1781. IEEE, 2007.

[16] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice. Position based visual servoing: keeping the object in the field of vision. In *IEEE International Conference on Robotics and Automation, 2002*, volume 2, pages 1624–1629. IEEE, 2002.