Frequency Domain Visual Servoing using Planar Contours

Visesh Chari Avinash Sharma Anoop Namboodiri C. V. Jawahar
Center for Visual Information Technology
International Institute Of Information Technology, Hyderabad, India
{visesh@research., avinash_s@research., anoop@, jawahar@}iiit.ac.in

Abstract

Fourier domain methods have had a long association with geometric vision. In this paper, we introduce Fourier domain methods into the field of visual servoing for the first time. We show how different properties of Fourier transforms may be used to address specific issues in traditional visual servoing methods, giving rise to algorithms that are more flexible. Specifically, we demonstrate how Fourier analysis may be used to obtain straight camera paths in the Cartesian space, do path following and correspondence-less visual servoing. Most importantly, by introducing Fourier techniques, we set a framework into which robust Fourier based geometry processing algorithms may be incorporated to address the various issues in servoing.

1 Introduction

The task of positioning a robot with respect to an object may be defined as the *constant manipulation of the position of a robot with respect to object (s) of the environment, while receiving and incorporating constant feedback about localization from its sensors.* The task of positioning is useful for various reasons. For example, the simple task of an automatically driven car requires the car to constantly monitor its position with respect to the road, and to constantly evaluate what turn to make and when; all these tasks amount to the *positioning* of the car with respect to various points on the road [17]. In any such task, a *sensor* is required to provide constant input about the environment, which is then coupled with *positioning algorithms* that determine and correct the position of the robot.

In this paper, we look at a positioning task called visual servoing. Visual servoing can be explained as the task of manipulating the motion of a robotic arm so that it moves from one position known as the start position to the desired or end position, with respect to a desired object (or target). Along this path, constant feedback in terms of *visual input* (through a camera) is coupled with servoing algorithms which confirm whether the robot is on the correct path, or corrects it when the robot goes astray. The nature of the path followed depends on the control law (objective function whose outcome determines the direction of motion of the robotic arm at each instant) designed, which in turn depends on the image features being utilized in the objective function for feedback. Visual servoing approaches model robot motion as an error minimization process [14, 18, 9, 4], which is a function of the pose of the camera and the features used for navigation. Error models employed in the past include image and pose errors [9], errors in homography [20] and moments [19], etc. The path the robot has to follow at any instant is directly derived as motion along the gradient of this pose-feature (or pose-error) space. The estimation of the direction of motion is usually done using a Jacobian which is a first order approximation of the gradient in the pose space at that particular point.

Had the error function been convex, this approach would have lead to a global minimization algorithm, which takes the robot manipulator or end-effector from the initial to final positions. However, the inherent non-linearity of perspective projection gives rise to problems such as local minima of the error function which means that there exists points in the pose-error space where the error is non-zero, yet the above algorithm would be incapable of giving a particular direction for the robot to proceed resulting in stagnation. Another popular problem is the non-straight Cartesian paths for the robot, during the minimization process. This means that when the resulting error is a difference in image coordinates (image points are considered as features) of an object captured in the initial and final poses, the error minimization ensures that the robot proceeds in a straight path in the *image space.* Because perspective projection is non-linear, this straight path in the image space results in a *non-straight* Cartesian path, which means the manipulator travels more than it needs to and runs the risk of reaching its joint limits in the process (joint limits are reached when the robot cannot stretch its arm beyond a point in order to reach a particular position in space). In addition, exact correspondence between points in the current and desired views need to be estimated for accurate convergence.

In order to counter these problems, research on different types of features and control laws has been carried out in

978-0-7695-3476-3/08 $25.00 © 2008 IEEE
DOI 10.1109/ICVGIP.2008.63

IEEE computer society

the past [16, 13, 20, 3]. However, of all the methods proposed, the traditional method of Image Based Visual Servoing (IBVS) [15] has proven stability and convergence for certain cases. Hence error functions defined in the image space can benefit from these results.

A popular class of higher order primitives that may be extracted from an image is Fourier descriptors. Fourier transform has seen wide use in the computer vision community for geometry related computations. Specifically, Fourier based methods have been used to estimate quantities such as the Fundamental matrix for registering cameras [12], and homography for registering images of contours [10]. Geometric computations based on Fourier transforms are robust to image noise, since each frequency has a collection of measurements from the entire image or contour. Recently, Fourier based techniques have also been applied for correspondence estimation [10]. Thus, visual servoing approaches set in a Fourier based framework have the potential of bringing these advantages into the servoing process. However, the restriction of Fourier based geometric computations to affine transformations [1] has been a bottleneck in applying them to visual servoing. In this paper, we resolve this problem in the context of visual servoing, and present an approach for servoing based on Fourier transforms.

Our contribution lies in choosing different properties of Fourier transforms to address different problems in traditional servoing.

- A correspondence-less visual servoing algorithm, built over the traditional IBVS framework, which is capable of positioning the robotic arm at a desired position *without* knowledge about explicit point-to-point correspondences between the object contour imaged in the initial and final views.

- Traditional IBVS algorithms result in a non-straight Cartesian path due to the non-linearity of perspective projection, mainly due to the effect of out-of-plane rotation. We show how a Fourier transform based method employing Taylor series can resolve this problem in the case of a 5 DOF robot.

- A control law employing weighted Fourier features, capable of following a user given path *while* minimizing the corresponding error funtion over time.

Currently, we consider servoing in the case of 1D Fourier transforms of planar object contours, since it simplifies the underlying geometric transformations to the extent that approximations provide reasonable solutions. One could potentially extend this approach to non-planar objects as well as 2D Fourier transform.

2 Related Work

Some of the initial works that inspired our approach was the application of frequency domain techniques to the estimation of multiple view geometric quantities like the homography [10, 11]. In [11], the authors propose the correspondence-less estimation of an affine transformation between planar contours imaged in two views as the estimation the transformation between their respective frequency transforms. This problem is then solved by imposing a rank constraint on a measurement matrix obtained from the respective Fourier coefficients. Additionally the correspondence between the two images is obtained by computing the shift between the Fourier components of the respective contours. This has been extended in [10], where the authors start from the solution provided by [11] and iteratively estimate the *projective* transformation between the input images of the planar contour, again with correspondence as the by-product.

The closest work to our approach is the work by Chaumette *et. al.* [19, 2], where the authors define a set of moments as features for the purposes of visual servoing. Moments are higher order equations in terms of the image coordinates, and are shown to be able to capture certain properties about the shape of the contour in consideration that is independent of the angle of view of the camera. These moments have nice properties like allowing the individual control of each of the 6 degrees of the pose of the camera. The representation of Fourier transforms as exponentially weighted image pixels is close to the representation of image moments. Indeed, this leads to several similarities like the linearization of projective transformations in order to decouple rotation and translation. However, unlike [19], our attempt is to *improve* IBVS by using Fourier based processing rather than proposing a novel control scheme for servoing. This in turn, allows us to make use of stability and convergence properties of IBVS. Again, for example, it is possible to combine the path following method with IBVS to get a control law that follows a given path while keeping features in the field of view.

3 Fourier Based Visual Servoing

Since in visual servoing, the basic aim is to get a direction to move the robot at every instant, the application of Fourier analysis seems promising. In this section, we show how the properties of Fourier representation of contours may be used to do three important tasks in visual servoing.

3.1 Correspondence-less Servoing

Image Based Visual Servoing (IBVS) defines the path followed by the manipulator as descent along a pose-image error function. This function is described as the sum of squared differences between the corresponding points in the initial and final views. This function, however, implicitly makes the assumption that correspondence between the two views has been established *apriori.* This is a problem when correspondence between views cannot be established easily or accurately. Correspondence-less visual servoing is particularly useful when the two views of the object are widely separated leading to inaccurate correspondences. Traditional IBVS cannot correct this error, hence there is a need to address this problem. We show now that in case of planar contours, this problem may be addressed by the use of frequency domain techniques [10, 11].

Let $X = \{x_i\}, Y = \{y_i\}$ and $X\_s = \{x\_s_i\}, Y\_s = \{y\_s_i\}$ be the x and y coordinates of a set of two contours that are images of a planar contour in 3D, imaged in two cameras. Let us assume that the correspondence between the two images is unknown, however, by creating $X,\ Y$ such that the points of the contours are either stacked in a clockwise direction or anticlockwise direction, we can reduce the correspondence estimation problem to an estimation of *shift* between the two contours [10]. We wish to be able to servo so that the camera moves from the first view to the second view in the absence of knowledge about this shift. Let $\mathcal{F}$ represent the 1D fourier transform. This change in shift comes up as a multiplicative factor in the Fourier transform

$$\mathcal{F}_k(X\_s) = \mathcal{F}_k(X)\ exp^{2\pi jk\lambda} \tag{1}$$

$$\mathcal{F}_k(Y\_s) = \mathcal{F}_k(Y)\ exp^{2\pi jk\lambda} \tag{2}$$

where $\mathcal{F}_k$ represents the k^{th} frequency component. When the views are related by an affine transformation, several methods are available to estimate the correspondence between contours [11]. But in the case where the contours are transformed by a more general *projective* transformation, these algorithms only give an *approximate* estimate of the correspondence. However, this approximation gives rise to a simple algorithm that can be used to perform servoing in a Fourier based framework.

We define correspondence less servoing as a two step process, when a generic projective transformation exists between the initial and desired images. In the first step, shift between the two contours is estimated using the algorithm of [11]. In the next step, IBVS is performed by using an estimation of this shift to generate the error vector [18]. As the servoing process converges the robot to the desired view, the two views (current and desired) approach each other, and in such a case, the transformation between the two planes is known to be affine [7], for which estimation of shift is accurate. The reasoning behind our approach is the following. Affine transformations are known to be good approximations of projective transformations [6], since effects of perspective projection get prominent only when the object under consideration is very close to the camera or the camera moves drastically [7]. Thus, estimation of shift assuming affine transformations is a close approximation to the actual shift present between the contours. Since IBVS is robust to such small errors in feature correspondence, the servoing is minimally affected because of this erroneous correspondence and so the robot moves in the correct direction. Eventually, when the views get close enough, the affine part of the transformation takes over and more accurate correspondences are obtained. The overall algorithm is summarized in Algorithm 1.

In practice, in comparison to the algorithms proposed in [11] for shift estimation, we found the following derivation to be more robust in estimating the correspondence under projective transformations.

$$\mathcal{F}_k(X\_s) = \mathcal{F}_k(X) * exp^{(2\pi ik\lambda/n)} \tag{3}$$

$$\log(\frac{\mathcal{F}_k(X\_s)}{\mathcal{F}_k(Y)}) = 2\pi ik\lambda/n \tag{4}$$

$$\lambda = \max_{\lambda} |(\mathcal{F}^{-1}\left(\frac{n}{2i\pi k}\log(\frac{\mathcal{F}_k(X\_s)}{\mathcal{F}_k(x)}\right))|, \tag{5}$$

which in words, means that we are trying to minimize the *maximum possible shift* between the two contours in consideration. This is particularly useful, when the contour does not have a random shape but has a specific pattern to it, like the one shown in Figure 1.

Algorithm 1 Algorithm for correspondence-less visual servoing.

```
Input: Initial and desired contour images.
Extract contour points, $X = (x_i),\ Y = (y_i)$ and $X\_s_k = (x\_s_i), Y\_s_k = (y\_s_i)$.
for $k = 1 \ldots \infty$ do
    Estimate shift $\lambda$ between the two contours using Equation ( 5).
    Construct error matrix using $\lambda$ [18].
    If contours are sufficiently close, exit.
end for
```

3.2 Straight Cartesian Paths for 5 DOF Servoing

In the previous section, we formulated a simple algorithm for correspondence-less servoing that can work even in scenarios where correspondence is difficult to obtain. In this section, we focus on the straight Cartesian trajectory problem, and propose a new algorithm that can achieve near-straight Cartesian paths in the case of a 5 DOF (degrees of freedom) visual servoing process. Since we achieve straightening of the path by a linearization in the rotation space, it becomes exceedingly complex to handle a 6 DOF robot with 3 degrees each for translation and rotation. Thus we consider only 2 degrees of rotational freedom in our current work.

Let the image of a 3D contour in the initial and desired views be represented by $\{X_i^2\}$ and $\{X_i^1\}$, where superscript denotes the frame number and the subscript denotes the

i^{th}point. Each point $X_i^k = \left[x_i^k \; y_i^k\right]$. Let us further assume that the two contours are related by a homography f.

In the case of planar contours, an affine transformation is comprised of the effects of 4 parameters of pose out of the existing 6 [7]. These are translation in x (t_x), y (t_y) and z (t_z), and rotation about the z axis (r_z). The *non-linearity* of the image motion - Cartesian motion relationship is primarily due to the remaining rotation parameters $(r_x,\, r_y)$. Our aim is thus to get an estimate of r_x or r_y, given the contours, since in a 5-DOF robot, the other pose parameters can be readily decoupled.

For the sake of simplicity, assume the two given views only differ in r_x in their respective camera poses. This means that the values of $(r_{8,}r_9)$ are the main parameters to be estimated in Equation (9). The case for r_y can be made on similar lines. Extension to a generic pose change is discussed later. The points in the two images are related by f which is an *infinite homography* [7].

$$X_i^2 = f(R, X_i^1) = \left[\begin{array}{ccc} f_x(R, X_i^1) & f_y(R, X_i^1) & 1 \end{array}\right]^\top \quad (6)$$

$$f_x(R, X_i^1) = x_i^2 = \frac{R_1 X_i^1}{R_3 X_i^1} \quad (7)$$

$$f_y(R, X_i^1) = y_i^2 = \frac{R_2 X_i^1}{R_3 X_i^1} \quad (8)$$

$$R = \left[\begin{array}{ccc} R_1 & R_2 & R_3 \end{array}\right]^\top = \left[\begin{array}{ccc} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{array}\right], \quad (9)$$

where R represents the rotation between the two views under consideration. We now perform a Taylor expansion of f with respect to R gives us a linear representation of the rotation, about $I_{3\times3}$.

Linearizing the right hand side of Equations (7 & 8) about the identity matrix, we get

$$f_x(R, X_i^1) = \frac{I_1 X_i^1}{I_3 X_i^3} + \frac{\partial f(I, X_i^1)}{\partial r_i}(r_i - i_i) \quad (10)$$

$$f_x(R, X_i^1) = (2 - r_9)x_i^1 - r_8 x_i^1 y_i^1 \quad (11)$$

$$f_y(R, X_i^1) = y_i^1 - r_8 y_i^1 y_i^1 + r_6 \quad (12)$$

where the values of R are set according to a rotation about x-axis. Notice that the major cause of worry is the parameter r_6, which is conveniently decoupled from the rest of the parameters, and present just as an addition term. This ensures that its effect can be removed by subtracting the DC component from the Fourier transform.

We now take the Fourier transformation of both sides of Equations (11 & 12), to obtain

$$\mathcal{F}(x_i^2)= \mathcal{F}(f_x(R, X_i^1)) = (2 - r_9)\mathcal{F}(x_i^1) - r_8\mathcal{F}(x_i^1) * \mathcal{F}(y_i^1) \quad (13)$$

$$\mathcal{F}(y_i^2) = \mathcal{F}(f_y(R, X_i^1)) = \mathcal{F}(y_i^1) - r_8\mathcal{F}(y_i^1) * \mathcal{F}(y_i^1) + \mathcal{F}(r_6) \quad (14)$$

where $*$ represents convolution.

The above equations expand into 2 equations *per* frequency of the Fourier transform. Since all the points share the same values of the rotation parameters, and since the Fourier parameters $(\mathcal{F}(X_i^j))$ are known to us, the estimation of the remaining parameters $(r_8,\, r_9)$ becomes a linear estimation problem. By collecting the parameters to be estimated in the above two equations into a matrix and inverting the sides of the equations, we get the following

$$\left[\begin{array}{cc} \mathcal{F}(x_i^1) & \mathcal{F}(x_i^1) * \mathcal{F}(y_i^1) \\ 0 & \mathcal{F}(y_i^1) * \mathcal{F}(y_i^1) \end{array}\right] \left[\begin{array}{c} (2 - r_9) \\ -r_8 \end{array}\right] = \left[\begin{array}{c} \mathcal{F}(x_i^2) \\ \mathcal{F}(y_i^2) - \mathcal{F}(y_i^1) \end{array}\right] \quad (15)$$

where $\mathcal{F}(r_6)$ is assumed to be cancelled out due to subtraction of the DC component. The above equation can be collected over all the frequencies except the base frequency to obtain an *over-determined* set of equations that can be solved in the least squares sense using SVD [8]. The method is summarized in algorithm 2

Algorithm 2 Simple iterative estimation of rotation, between contours.

```
Input: Two contours, $(X_i^1)$ and $(X_i)$.
$(X_i^3) = (X_i^2)$
while convergence is not achieved upto acceptable error
do
    Compute $\mathcal{F}(X_i^3)$ and $\mathcal{F}(X_i^2)$. Discard DC.
    Build error matrix (Equations 13, 14, 15).
    Minimize the error using SVD [8].
    Rotate $(X_i^3)$ to obtain the new contour. Assign it to $(X_i^3)$ for the next iteration.
    Collect the rotation matrix produced.
end while
Composition of all the collected rotation matrices provide the entire transformation between $(X_i^2)$ and $(X_i^1)$.
```

Extension to general pose change The challenge now is to estimate r_x or r_y when the remaining pose parameters $(t_x,\, t_y,\, t_z,\, r_z)$ are not assumed zero. Since these parameters form an affine transformation, results from [1] may be leveraged to our advantage. Specifically, when a contour undergoes an affine transformation $H = \left[\begin{array}{cc} A & b \\ 0 & 1 \end{array}\right]$, the corresponding Fourier components undergo the same transformation. Since $(t_x$,$t_y)$ primarily contribute to the DC component and (t_z) to scale, higher frequencies are suitable to estimate (r_x) or (r_y) robustly. Now consider an intermediate view X_i^3, which differs from the current view (X_i^1) in a rotation about the x or y axis. Thus Fourier transform of the relationship between X_i^3 and X_i^2 may be described by an affine transformation.

$$\mathcal{F}(X_i^2) = \lambda(A\mathcal{F}(X_i^3) + \mathcal{F}(b)) \quad (16)$$

The unknown scale factor λ approximates the change t_z, and may be described as the ratio of areas of the contour in the two images [4]. Expanding the relationship between

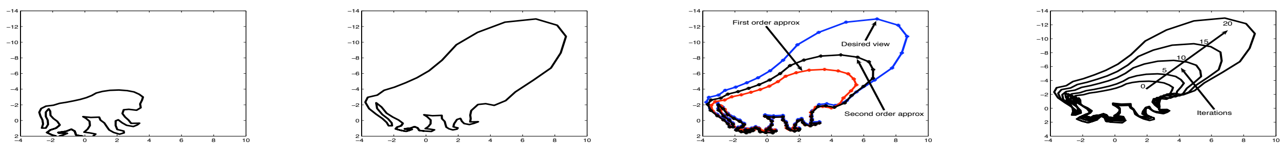

Figure 1. (Left-Right): (a) Current view (b) Desired view (c) First and second order Taylor approximations of the rotation about x-axis. (d) Different stages of the iterative minimization process.

X_i^3 and X_i^1 in Equation (6) using the Fourier relationship derived earlier and ignoring the base frequency to eliminate r_6, we get a matrix along the lines of Equation (15) as

$$\mathcal{F}(X_i^2) = \lambda(\begin{bmatrix} a_{11}(2-r_9)\mathcal{F}(x_i^1) + a_{12}\mathcal{F}(y_i^1) - r_8 a_{11} \\ \mathcal{F}(x_i^1) * \mathcal{F}(y_i^1) - r_8 a_{12}\mathcal{F}(y_i^1) * \mathcal{F}(y_i^1) \\ \\ a_{21}(2-r_9)\mathcal{F}(x_i^1) + a_{22}\mathcal{F}(y_i^1) - r_8 a_{21} \\ \mathcal{F}(x_i^1) * \mathcal{F}(y_i^1) - r_8 a_{22}\mathcal{F}(y_i^1) * \mathcal{F}(y_i^1) \end{bmatrix}) \tag{17}$$

These set of equations in 6 unknowns, 4 of A and 2 of R can be solved in the least squares sense using the equations resulting from the various frequencies in the Fourier transform. The parameter λ may be eliminated by imposing the condition that A and R are rotation matrices and hence have unit norm. Once the rotation values are estimated, the translational degrees of freedom may be controlled separately leading to a decoupled system. The entire method is summarized in Algorithm 2, with the only change that step 4 is replaced by Equation (17).

3.3 Path Following

Having considered two problems with traditional visual servoing in the previous sections, we now turn our attention to a novel problem, that of *altering the path* of minimization in the visual servoing process.

We show how a minimization of the weighted Fourier error can be used as a method to follow *different paths* in the minimization process, while the robotic manipulator goes from the *same* initial position to the *same* final position. Thus, such an algorithm can take as input, say the output of a path planning algorithm that plans the path of a robot through an environment taking into consideration the obstacles present along the way. Our algorithm will then *follow* the obtained path, while minimizing the error between current and desired frames.

We start the derivation of our control law by first observing a snapshot of the equations involved in traditional IBVS, which is shown below, where $e = s - s^*$ s, s^* being the feature vectors in initial and final positions.

$$\dot{e} = \frac{\partial e}{\partial p}\frac{\partial p}{\partial t} \tag{18}$$

$$\frac{\partial p}{\partial t} = J\dot{e}, J = \left(\frac{\partial e}{\partial p}\right)^{+} \tag{19}$$

where $(e, \dot{e})$ represent the image error and the rate of decrease of image error, respectively. p represents the pose of the robot manipulator, while J represents the Jacobian that ultimately decides the direction of movement at every step of the convergence.

In traditional image based visual servoing (IBVS), the control law is specified in terms of the image error $e = s - s^*$, and the Jacobian J_s (Equation (19)). The velocity screw of the robot end-effector at any time is then given by

$$J_s = \begin{bmatrix} J_1 & \ldots & J_n \end{bmatrix}^{\top} \tag{20}$$

$$J_i = \begin{bmatrix} 1/Z_i & 0 & -v_i/Z_i & -u_i v_i & 1+u_i^2 & -v_i \\ 0 & 1/Z_i & -u_i/Z_i & -(1+v_i^2) & u_i v_i & u_i \end{bmatrix} \tag{21}$$

$$v = J^{+} e \tag{22}$$

We now define the Fourier error to be

$$e_{\mathcal{F}} = \mathcal{F}(e) = \mathcal{F}(s) - \mathcal{F}(s^*), \tag{23}$$

where $\mathcal{F}(s)$ is a concatenation of the Fourier transforms of the x and y coordinates. Since Fourier transforms are linear and invertible, the Jacobian of the Fourier error can be computed in a manner similar to the traditional IBVS error. It turns out to be

$$J_{\mathcal{F}} = \frac{\partial F}{\partial s} J_s \tag{24}$$

$$J_{\mathcal{F}}^{+} = J_s^{+}\left(\frac{\partial \mathcal{F}}{\partial s}^{-1}\right) \tag{25}$$

$$= J_s^{+}\frac{\partial(\mathcal{F}^{-1})}{\partial s}, \tag{26}$$

where $\mathcal{F}^{-1}$ denotes the inverse Fourier transform. By defining the Fourier error and related control law, we have ensured that the feature vector used for the minimization process is the Fourier descriptor of the image points, which may be a contour. The advantage of Fourier descriptors is that each frequency component captures *global* properties about the contour in consideration, unlike image points, which are local in nature. Since we have already seen earlier that Fourier components tend to capture the geometric properties of the transformation between initial and final views [11, 10], we use this concept to argue that changing each of these components has a different effect on the overall image error of traditional IBVS, and hence the minimization process.

Thus, to control this effect, we now introduce a *weight* matrix W, that weights each component of the Fourier error. Thus our new error model can now be defined as

$$\begin{aligned} e_{\mathcal{F}}^{W} &= W e_{\mathcal{F}} && (27) \\ W &= \text{diag}\left[\begin{array}{ccc} w_1 & \dots & w_n \end{array}\right] && (28) \\ v &= J_s^{+} \frac{\partial \mathcal{F}^{-1}}{\partial s}(W e_{\mathcal{F}}) && (29) \end{aligned}$$

The velocity screw at each iteration is then given by Equation (29). Since each Fourier component is composed of contributions from all the points in the image, weighting the Fourier components has a global effect on the error minimization. This allows IBVS to follow different paths within the same minimization framework.

To demonstrate the application of the above control law, assume that we have a sequence of relative pose estimates $\{\Delta p_i\}$ with respect to the destination frame available to us. These estimates may be the output of a constrained optimization problem used in path planning [16]. In order to ensure that the servoing algorithm makes the robot pass through these given poses, we need to find the weight matrix W that will guide the robot appropriately. In order to do this, we observe that in Equation (29), the left hand side represents the screw velocity, which is in turn defined as change in pose over time (between successive frames, this takes the value of one instant). Thus if we define $v = \Delta p_i$ in the above equation, the inversion of the right hand side will give the *difference* between the estimated pose change in the current instant and the *desired* pose change, represented by Δp_i . We can now define the desired weight matrix as the value that *minimizes* this difference as

$$W = \min_{W}(-J_F^{+} W e_{\mathcal{F}} - \Delta p_i)^2, \quad (30)$$

where the negative sign for error indicates the Jacobian at the desired pose. Reordering the terms in the equation by substituting W for a vector of its diagonal elements and $e_{\mathcal{F}}$ for a matrix with elements in its diagonal, we get

$$\begin{aligned} W &= \min_{W}(-J_{\mathcal{F}}^{+} e_{\mathcal{F}}^{*} W^{*} - \Delta p_i)^2 && (31) \\ W^{*} &= (-J_{\mathcal{F}}^{+} e_{\mathcal{F}}^{*})^{-1} * \Delta p_i && (32) \\ W^{*} &= -e_{\mathcal{F}}^{*-1} J_{\mathcal{F}} \Delta p_i && (33) \end{aligned}$$

As can be seen, although the relationship between pose change and the different components of the Fourier transform are not completely known, nevertheless we are able to control the path of the robot by weight individual components. This equation holds as well near the desired pose as the approximation of the Jacobian for IBVS. The entire method is summarized in Algorithm .

Algorithm 3 Path following using weighted Fourier servoing.

```
Input: initial and desired images for the servoing.
Input: A list of poses ($\Delta p_i$) to achieve.
Append desired pose $\Delta p_{desired}$ to the given list.
Set k = 1
while convergence to desired pose not achieved within bounds do
  Compute $W$ to servo to $\Delta p_k$ (Equation 31).
  Obtain direction (Equation 29).
  Servo.
  if $\Delta p_k$ achieved within error bounds then
    k = k + 1
  end if
end while
```

4 Results and Analysis

In the next few paragraphs, we present the results of our three algorithms on simulation data. This kind of simulation based experiments is actually a standard in the visual servoing literature [5], with simulation toolkits in Matlab being released by some of the pioneers in the field.

4.1 Straight Cartesian Paths for 5 DOF Servoing

Figure 1 shows the setup in which the current and desired views are separated by 10 degrees of rotation about the x-axis and z-axis. The decoupling involves minimizing the error in rotational components of the views first, followed by translation. Each iteration of the minimization process consists of finding the least square solution for the angle (r_x) and minimizing the rotational error by rotating the camera appropriately. Since the initial and final views are widely separated, at first this minimization is approximate. However, as the camera moves nearer to the desired pose, the relationship between the two views moves towards an affine transformation, which can be handled better by Fourier transforms. It is interesting to note that each frequency component gives 2 equations for a total of $2(n-1)$ equations barring the base frequency. Had correspondences been unknown, we would still have $(n-1)$ equations following the idea in Section 3.1 which opens up the possibility of pose computation without correspondence in planar scenes, useful for PBVS.

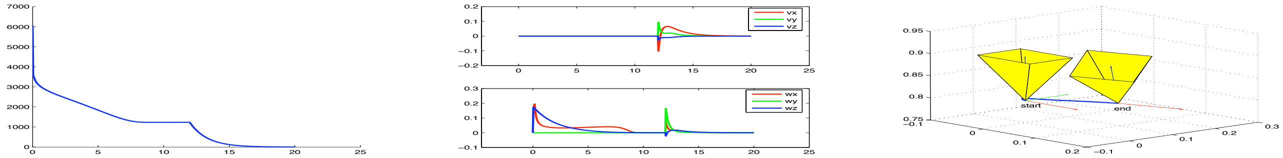

Figure 2. Cartesian straight path for 5 DOF (Left-Right): (a) Image error increases during rotation. (b) Observe how the rotation is corrected first and then the translation. (c) The slight error towards the end of the path is due to the approximations used in the formulation.

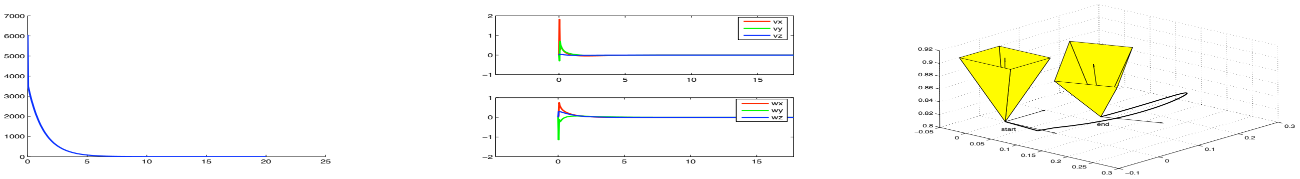

Figure 3. Image based visual servoing (Left-Right): (a) Image error decreases exponentially. (b) Observe how the corrections in rotation and translation lead to the path of the robot being curved in (c).

Figure 2 shows the visual servoing trajectory after plugging in the decoupling solution, above the normal visual servoing trajectory. Note how the decoupling results in a straight path in Cartesian coordinates when normal IBVS would result in a curved trajectory.

The main drawback of this method is the Taylor approximation that needs to be leads to small errors in the initial rotation error minimization. However, we observe that the minimization process converges even for huge perspective distortions (Figure 1), and hence believe this to be a promising result in spite of the minute error.

4.2 Path Following

Figure 4 shows the results of the path following approach. Figure 4(a) shows the camera trajectory during normal visual servoing, while Figure 4(b) the desired camera trajectory. Notice how, even with the linear approximation, the actual camera trajectory 4(c) closely mimics the desired trajectory. This suggests that the Fourier-pose space curves slowly around the desired pose, and thus the linearization approximates Fourier error for a large region around the desired pose.

The path following approach is particularly useful for tasks like local minima avoidance in IBVS and for keeping features in field of view. However, since explicit relationships between each of the Fourier components and the elements of camera pose (rotation, translation) is difficult to analyze, only approximate relationships for both the tasks can be devised. However, the concept of weights allow for a smooth camera trajectory that can satisfy more than one constraint. This is a useful contribution considering joint and velocity limits of robots.

4.3 Correspondence-less Servoing

The results for correspondence-less visual servoing are shown in Figure 5. In this experiment, we have considered two contours in initial and desired views with 100 points sampled in each view. The correspondence misalignment is around 20 pixels. As expected, the actual shift has little consequence on the minimization procedure. The increasing accuracy in correspondence results in many artifacts in the minimization procedure, where the correspondence between the current and desired views shifts by a particular number. For example, at the start of the IBVS iteration, the correspondence is estimated as a 15 pixel shift, which gets refined to 20 over time.

5 Conclusions

In this paper, we have presented several reasons to set visual servoing algorithms in a Fourier based framework, like correspondence-less servoing, straight Cartesian path servoing and path following. Fourier based approaches are particularly attractive since recent results in geometric vision [12] could be easily leveraged to advance visual servoing.

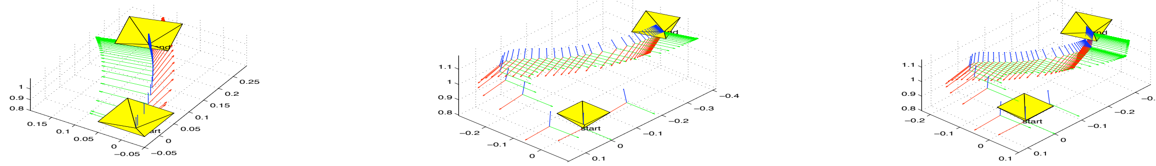

Figure 4. Path following (Left-Right): (a) Image based visual servoing (IBVS). (b) Specified path to be followed, and (c) Path of the end-effector following trajectory. Observe how closely the specified trajectory is followed.

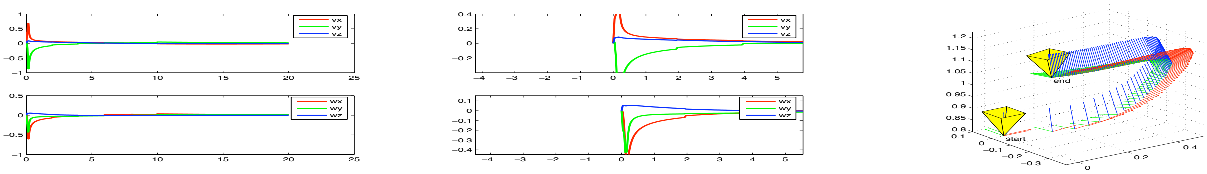

Figure 5. Correspondence-less visual servoing (Left-Right):(a) Velocity screw in the servoing process. Notice the slight bumps in the screw (magnified in (b)). (c) Corresponding camera trajectory that is slightly more skewed than traditional visual servoing.

References

[1] R. N. Bracewell. *Fourier Analysis and Imaging*. Kluwer Academic/Plenum Publishers, 2003.

[2] F. Chaumette. Image moments: a general and useful set of features for visual servoing. *IEEE Trans. on Robotics*, 20(4):713–723, August 2004.

[3] C. Collewet and F. Chaumette. A contour approach for image-based control of objects with complex shape. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'00*, volume 1, pages 751–756, Takamatsu, Japon, November 2000.

[4] P. Corck and S. Hutchinson. A new partitioned approach to image-based visual servoing. *IEEE Transactions on Robotics and Automation*, 14(4):507–515, Aug 2001.

[5] P. I. Corke. A computer tool for simulation and analysis: the robotics toolbox for matlab. In *Proc. National Conf. Australian Robot Association*, pages 319–330, 1995.

[6] A. Criminisi, I. Reid, and A. Zisserman. A plane measuring device. *Image and Vision Computing*, 17(8):625–634, 1999.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003.

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[9] S. Hutchinson, G. Hager, and Cork. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 17:18–27, 1996.

[10] P. K. Jain and C. V. Jawahar. Homography estimation from planar contours. *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, 2006.

[11] S. Kuthirummal, C. V. Jawahar, and P. J. Narayanan. Fourier domain representation of planar curves for recognition in multiple views. *Pattern Recognition*, 2004.

[12] S. Lehmann, A. P. Bradley, I. V. L. Clarkson, J. Williams, and P. J. Kootsookos. Correspondence-free determination of the affine fundamental matrix. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[13] R. Mahony, P. Corke, and F. Chaumette. Choice of image features for depth-axis control in image-based visual servo control. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'02*, volume 1, pages 390–395, Lausanne, Switzerland, October 2002.

[14] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *IEEE Trans. on Robotics and Automation*, 1999.

[15] E. Malis and P. Rives. Robustness of image-based visual servoing with respect to depth distribution errors. In *IEEE Int. Conf. on Robotics and Automation, ICRA'03*, volume 1, pages 1056–1061, Taipei, Taiwan, Sept. 2003.

[16] Y. Mezouar and F. Chaumette. Path planning for robust image-based control. *IEEE Trans. on Robotics and Automation*, 18(4):534–549, August 2002.

[17] C. Pradalier, J. Hermosillo, C. Koike, C. Braillon, P. Bessière, and C. Laugier. The cycab: a car-like robot navigating autonomously and safely among pedestrians. *Robotics and Autonomous Systems*, 50(1):51–68, 2005.

[18] P. Rives, B. Espiau, and F. Chaumette. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 1992.

[19] O. Tahri and F. Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Transactions on Robotics*, 21(6):1116–1127, December 2005.

[20] M. Vargas and E. Malis. Visual servoing based on an analytical homography decomposition. *CDC-EDC '05, 44th IEEE Conference on Decision and Control*, pages 5379– 5384, 2005.