Improvement to the Minimization of Hybrid Error Functions for Pose Alignment

A. H. Abdul Hafez[1,2]
[2] Dept. of Computer Science and Engineering
University College of Engineering, Osmania University
Hyderabad-500007, India
Email: hafezsyr@ieee.org

C. V. Jawahar[1]
[1] Center for Visual Information Technology
International Institute of Information Technology
Gachibowli, Hyderabad-500032, India
Email: jawahar@iiit.ac.in

***Abstract*— Many problems in computer vision such as pose recovery and structure estimation are formulated as a minimization process. These problems vary in the use of image measurements directly or using them to extract 3D cues in the minimization process. Hybrid methods have the advantage of combining the 2D and 3D visual information to improve the performance over the above two methods. In this paper, we present a new formulation for minimizing a class of hybrid error functions. This is done by using 2D information from the image space and 3D information from the Cartesian space in one error function. Applications to visual servoing and image alignment problems are presented. The positioning task of a robot arm has been formulated as a minimization problem. Gradient decent as a first order approximation and Gauss-Newton as a second order approximation are considered in this paper. Simulation results show, comparing with 2 1/2 D hybrid method, that these two methods provide an efficient solution to the features visibility problems and the camera trajectory in the Cartesian space.**

I. Introduction

Minimization techniques play an essential role in solving a wide range of computer vision problems. Examples of these problems are camera pose recovery, motion and structure from video, visual servoing, image registration, augmented reality, tracking, etc. Most of the solutions to these problems optimize an appropriate objective function of the position. This function may be defined in the image (2D) space or in the world and camera (3D) space.

A large number of the computer vision applications estimate the camera position as a solution to the Least Squares Minimization (LSM) problem. Solutions based on LSM formulation are known to be optimal when independent and Gaussian measurement errors are assumed. One can distinguish between linear least-square minimization (LLSM) and non-linear least-square minimization (NLSM) based on the linearity or non-linearity of the cost function to be minimized.

In the case of linear cost functions, LLSM is effective and a closed form solution can be obtained. An example is the problem of 3D alignment by a rigid transformation. Most of the cost functions that use 3D features for pose alignment such as 3D points and pose information are linear with respect to the pose error. Therefore, LSM can produce an optimal closed form solution to such problems. However, many algorithms try to get an iterative solution from LLSM for the purpose of enforcing constraints like certain control trajectories or to add robustness.

In contrast, minimizing a non-linear function falls in the NLSM class. However, this minimization cannot be solved in closed form and requires iterative solutions to the optimization problem as an intermediate step in solving a larger problem. Cost functions from the 2D image space, such as the reprojection error, are highly non-linear and lead to solutions from NLSM. Usually, these solutions require an initial camera pose estimate. In applications such as visual servoing, a function of the initial pose estimate is already available.

The error function to be minimized may be defined in the image space as function of the reprojection error. The reprojection error is the squared distance between the projection of the 3D point represented in the current pose and their 2D coordinates measured in the desired pose. Examples about this error function are in 2D tracking applications. Another error function that is usually minimized is the distance between the current pose and the desired one as defined in the special Euclidean group $SE(3) = R^3 \times SO(3)$. This error function is estimated from the object pose in both desired and current camera frame using a priori knowledge about the scene, like position-based visual servoing, registration and augmented reality. Table I summarizes a comparison between 2D and 3D visual information and algorithms from different aspects.

In this paper, we propose a hybrid (2D-3D) cost function that contains information from the 2D image space and the 3D pose space. The closest work to ours is the error function that was proposed in the visual servoing literature [1], [2]. In [1], the error function was defined as a 6-vector. The 3-vector that contains the 2D visual information from the image space is used to recover the position of the camera, where the 3-vector that contains the 3D visual information from the pose space is used to recover the orientation of the camera. In contrast, our method uses each of the image space information and pose space information to recover the full camera pose information, both position and orientation. Both 2D and 3D errors are concatenated in a 12-vector and minimized together. In such a minimization method, the minimization process search for a solution that minimizes both the 2D error from the image space and the 3D error from the pose space together.

TABLE I

A brief comparison between 2D and 3D visual features in optimization process.

2D Features from the Image Space	3D Features from the Cartesian Space
Information that are extracted from image measurements are usually called 2D information. This visual information is represented using features such as points, lines, region of interest, contours, etc.	Visual information that represent the pose and/or the depth are called 3D information. This is represented by pose (position and orientation), 3D lines, 3D points, etc.
Using 2D visual features, error functions like pose and structure errors can be estimated using iterative non-linear minimization methods.A linearization method is usually needed for efficient numerical solution.	A linear relationship between these 3D features and the estimated parameters usually exist. A closed form solution using linear minimization process can often be found
The 2D visual features are extracted from the image and used directly in the minimization process.	The 3D information are obtained by processing an information extracted from the image itself along with the *a priori* knowledge about the camera.
Using 2D features from the image space often assume that the camera is internally calibrated.In case of estimating 2D parameters like homography, the camera calibration is not required.	The camera calibration is necessary for using any 3D features in the estimation process. The image is the only source for the measurements. Obtaining 3D information using triangulation needs camera calibration.

II. Minimization Approaches for Different Computer Vision applications

A. Augmented Reality Applications

Vision-based augmented reality system is an attractive interface for various applications such as video games, architecture, interior design, etc. Usually the focus is on the registration techniques that allow alignment of real and virtual worlds using images acquired in real-time by a moving camera. In such systems AR is mainly a pose (or viewpoint) computation issue. The problem is mainly addressed as the pose computation problem [3].

The most common methods, suitable for AR applications, to compute the pose, rely on indoor and outdoor fiducial markers. In the related computer vision literature, geometric features considered for the estimation are often points, segments, straight lines, contours or points on the contours conics, cylindrical objects or a combination of these different features [4]. Another important issue is the registration problem. Purely geometric numerical and/or iterative [5] approaches may be considered. Linear approaches use least-squares method to estimate the pose. Full-scale non-linear optimization techniques consist of minimizing the error between the observation and the forward projection of the model. In this case, minimization is handled using numerical iterative algorithms such as Newton-Raphson or Levenberg-Marquardt. The main advantage of these approaches are their accuracy. The main drawback is that they may be subject to local minima and, worse, divergence if initialization is not properly done.

These 2D-3D registration techniques rely on the use of a 3D model of the tracked objects. When such 3D models are not available, other approaches are required. If the full sequence is available *a priori* (such as for post-production applications), bundle adjustment techniques are considered. Bundle adjustment techniques involve estimating jointly optimal values for the scene structure and camera parameters using non-linear minimization techniques, which differ based on the parametrization.

B. Image Registration and Alignment

The registration of 3D objects is an important problem in computer vision and especially in medical imaging and Georegistration. It arises when data acquired by different sensors, at different places, and/or different time. Under the rigidness assumption of the objects to be registered, the problem is to recover the six parameters of a rigid transformation [6]. Whereas 2D images taken at different times do not necessarily represent the same spatial cross section, and do not contain the information necessary for correcting a misalignment, true 3D image acquisition makes the full six degree of freedom registration possible [7]. A general approach to image registration has three steps:

1) definition of common features found in all images
2) formulation of a transformation function to align image coordinate systems
3) reformatting of one (or more) images to bring datasets into alignment

A number of methods already exist [7], which explicitly minimize an energy function derived with respect to the parameters of the six degrees of freedom transformation. Research still continues on methods such as those avoiding the local minima and ensuring the convergence. Figure 1 shows an example about image alignment. In this figure two images of a face are given (Figs. 1(a,b)) along with the depth maps for each face model (Figs 1(c,d)). Using selected features from the image and its given map, the pose of each face is estimated iteratively. Here, 2D features or 3D features can be used.

C. Pose Alignment by Visual Servoing

Visual servoing schemes use one or more cameras along with computer vision algorithms to control the position of a robot arm or mobile robot with respect to an object or a set of feature of the object to be manipulated. It is used in a wide range of applications such as robot navigation, lane tracking by vehicles, and industrial manipulation.

The essence of visual servoing is to move the concerned object from the current pose to a desired pose given a current and desired images. This is essentially obtained by minimization of an error function. Visual features are extracted from the two images and used to formulate an error function between the current pose and the desired one. The role of the minimization process is to regulate this error function. The image features can be used directly in the definition of the

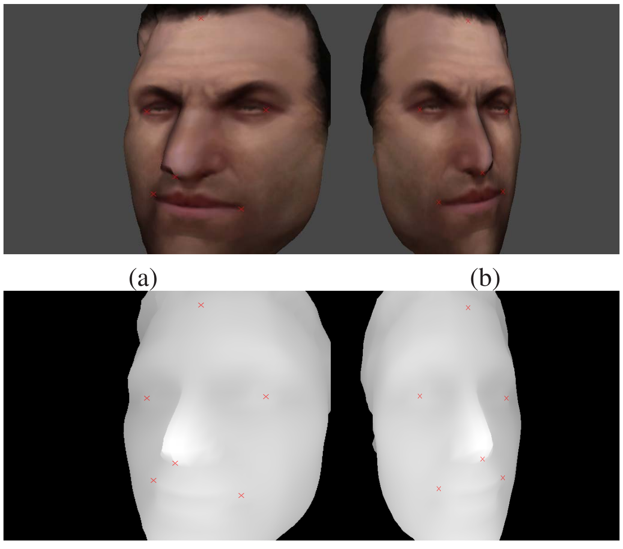


Fig. 1. Example of two images of a face from different poses in (a) and (b). The depth map corresponds to each image is given in (c) and (d). Features may be selected from the two images and matched to its depth from the depth map.

error function. This leads to a formulation of the 2D error function that is minimized in the image space. These image features may also be used to formulate an error function in the pose space such as the error between the position parameters and the error between the rotations parameters.

Another application of minimization process in visual servoing is tracking the visual features either in the image space or in the pose space. This is usually done by recovering the camera motion in the 3D Cartesian space or the motion model in the 2D image space.

III. Camera Pose Recovery Task as a Minimization Process

The camera pose recovery task is to estimate the desired camera P^* starting from an initial camera pose $P \in R^3 \times SO(3)$. In other words, the problem is to minimize an error vector $e(s)$ of visual features $s(P)$ by finding a vector ΔP that minimize a cost function $E(s(P))$. Viewing the problem as a nonlinear least squares minimization allows us to formulate the following cost function

$$E(s(P)) = \frac{1}{2}(s(P) - s(P^*))^T(s(P) - s(P^*)). \qquad (1)$$

In the remaining of this section, two minimization methods will be reviewed. The first one is based on the first order Taylor series approximation of the cost function. This method is called the gradient decent minimization. The second method is based on the second order Taylor series approximation of the cost function. This is called the Newton minimization.

A. Gradient Decent Minimization

Here, an approximation, using the first order derivative, of the above cost function is evaluated at the desired pose P^*. The cost function given in (1) can be written as

$$E(s(P^*)) \approx E(s(P)) + \frac{\partial E(s(P))}{\partial P}\Delta P. \qquad (2)$$

The gradient of the cost function $\frac{\partial E(s(P))}{\partial P}$ is given as

$$\frac{\partial E(s(P))}{\partial P} = (s(P) - s(P^*))^T \, \frac{\partial s(P)}{\partial P} = e(s)^T J(P), \qquad (3)$$

where $e(s) = s(P) - s(P^*)$. The criteria in gradient decent minimization is to move in the direction opposite to the gradient. Indeed, the required change in the pose is

$$V = \Delta P = -\lambda J^T(P)e(s), \qquad (4)$$

where λ is a positive constant parameter that defines the step size of the minimization process.

This method is known as the Jacobian transpose method in the robot control literature and was used in [8], [9] and recently restated in [10]. However, gradient decent methods are known to have a slow and linear convergence rate.

B. Newton Minimization

Here, an approximation using the second order derivative of the cost function is evaluated at the desired pose P^*. The cost function given in (1) can be written as

$$E(s(P^*)) \approx E(s(P)) + \frac{\partial E(s(P))}{\partial P}\Delta P + \frac{1}{2}\Delta P^T \frac{\partial^2 E(s(P))}{\partial P^2}\Delta P. \qquad (5)$$

The first order term $\frac{\partial E(s(P))}{\partial P}$ is as given in (3), while the second order term $\frac{\partial^2 E(s(P))}{\partial P^2}$ is given by

$$\frac{\partial^2 E(s(P))}{\partial P^2} = J^T(P)J(P) + \sum_{k=0}^{n} H_k(P)e_k(s). \qquad (6)$$

The matrix H_k is the Hessian matrix of the function $e_k(s)$. In Gauss-Newton minimization, the second order derivative is approximated by

$$\frac{\partial^2 E(s(P))}{\partial P^2} = J^T(P)J(P). \qquad (7)$$

Indeed, the required change in the pose is

$$V = \Delta P = -\lambda J^+(P)e(s), \qquad (8)$$

where the matrix

$$J^+(P) = \left((J^T(P)J(P)\right)^{-1} J^T(P)$$

is the pseudo-inverse of the matrix J. This method is known as the Jacobian Pseudo-inverse method and widely used in the robot control and visual servoing [11].

One may note that this method can be obtained form the gradient decent method. In this case, the error vector itself $e(s)$ is used in the minimization process instead of the cost function $E(s(P))$.

IV. Minimizing a Hybrid Error Function

Based on the type of the visual features $s(P^*)$ used in the minimization process, the cost function defined in (1) varies from $E_{2D}(P)$ for 2D visual features from image space to $E_{3D}(P)$ for 3D visual features from the Cartesian space.

If we consider the 2D coordinates of a set of image points as features, the vector $s(P)$ becomes $s_{2D}(P) = [x_1, y_1, \dots, x_N, y_N]^T$ while the desired vector $s(P^*)$ is $s_{2D}(P^*) = [x_1^*, y_1^*, \dots, x_N^*, y_N^*]^T$. Indeed, Equation (1) is rewritten as

$$E_{2D}(s(P)) = \frac{1}{2} e_i(P)^T e_i(P), \tag{9}$$

where

$$e_i(P) = s_{2D}(P) - s_{2D}(P^*). \tag{10}$$

In contrast, 3D visual features such as the position and orientation can be part of the feature vector $s_{3D}(P) = [T_x, T_y, T_z, u\theta]^T$. The desired features are $s_{3D}(P^*) = 0_{(6\times1)}$. Similarly to (9) and (10), we can write

$$E_{3D}(s(P)) = \frac{1}{2} e_p(P)^T e_p(P), \tag{11}$$

where

$$e_p(P) = s_{3D}(P) - s_{3D}(P^*) \tag{12}$$

It can be proved that minimizing either $E_{3D}(s(P))$ or $E_{2D}(s(P))$ results in equivalent results. In other words, $E_{3D}(s(P^*)) = E_{2D}(s(P^*)) = 0$.

Lemma 1: The cost function $E_{3D}(s(P)) = 0$ if and only if the cost fun$E_{2D}(s(P)) = 0$.

Proof: Let $M = [X, Y, Z]^T$ be a 3D point in the current camera frame and $m = [x, y, 1]$ is its perspective projection into image I. The image point m is given as $M = Zm$. Similarly, $M^* = Z^* m^*$ are the correspondence points in the desired camera frame. The cost function in the image space $E_{2D}(s(P)) = 0$ when $e_i(P) = m_i - m_i^*$, while the cost function in the 3D Cartesian space $E_{3D}(s(P)) = 0$ when $M = M^*$.

Proof of the **IF** condition:
The 3D point in the current frame relates to its correspondence in the desired one by the R and T as

$$M = RM^* + T \quad \Rightarrow \quad Zm = RZ^* m^* + T \tag{13}$$

$$m = m^*,\ Z = Z^* \quad \Rightarrow \quad R = I,\ T = 0_{(3\times1)} \tag{14}$$

Here, $T = [T_x, T_y, T_z]^T = 0_{(3\times1)}$ and $R = I_3$. Here, R is the matrix of the rotation represented by the axis u and angle θ.

Proof of the **ONLY IF** condition:
Given that $m = m^*$, we have

1) Both camera frames are identical, In other words $T = [T_x, T_y, T_z]^T = 0_{(3\times1)}$ and $R = I_3$.
2) The error is a translation along the insight line passing from the point m_i.
3) The error is a rotation about the insight line passing from the point m_i.
4) Both cases 2 and 3 hold.

In fact, the cases 2-4 may hold by considering a single image point $i = 1$, but they are totally incorrect if we consider more than one point $i \geq 2$. ■

Unfortunately, minimizing the cost function $E_{3D}(s(P))$ has a contradicted behavior with minimizing $E_{2D}(s(P))$. Some of the advantages and drawbacks of minimizing each of these two function has been summarized in Table I. Hybrid methods aim at reducing the undesirability while keeping maximizing the advantages of each of the two function minimization.

Let us define a hybrid cost function as the weighted sum of the two $E_{3D}(s(P))$ and $E_{2D}(s(P))$ function as follows

$$E_h(s(P)) = \lambda_1\ E_{2D}(s(P)) + \lambda_2\ E_{3D}(s(P)). \tag{15}$$

Here, λ_1 and λ_2 are positive scalar factors. It can be easily shown, using *Lemma 1*, that the hybrid function given in (15) is nullified only when $P = P^*$. The two constant λ_1 and λ_2 play the role of the step size of the minimization process in addition to the integration ratio between 2D and 3D spaces. While minimizing this cost function, the process searches for a solutions that reduce the value of the two individual functions $E_{3D}(s(P))$ and $E_{2D}(s(P))$. In the following two subsections we will show how the minimization methods presented in Section III can be used to minimize this hybrid cost function

A. Gradient Decent Method

Consider a positioning task to be achieved using the gradient decent minimization method. The hybrid cost function given in (1) will be rewritten as

$$E_h(s(P)) = \frac{\lambda_1}{2}\ E_{2D}(s(P)) + \frac{\lambda_2}{2}\ E_{3D}(s(P)), \tag{16}$$

where the division by 2 is useful to simplify the derivation process. The gradient vector of this cost function is given as

$$\begin{aligned}\frac{\partial E_h(P)}{\partial P} &= \frac{\lambda_1}{2} \frac{\partial\Big(e_i(P)^T e_i(P)\Big)}{\partial P} + \frac{\lambda_2}{2} \frac{\partial\Big(e_p(P)^T e_p(P)\Big)}{\partial P}, \\ &= \lambda_1 e_i(P)^T \frac{\partial\Big(e_i(P)\Big)}{\partial P} + \lambda_2 e_p(P)^T \frac{\partial\Big(e_p(P)\Big)}{\partial P}, \\ &= \lambda_1 e_i(P)^T J_i(P) + \lambda_2 e_p(P)^T J_p(P). \end{aligned} \tag{17}$$

The matrix $J_i(P) = \frac{\partial(e_i(P))}{\partial P}$ is the image Jacobian matrix, and the other matrix $J_p(P) = \frac{\partial(e_p(P))}{\partial P}$ is the Cartesian Jacobian matrix. To compute the change in the pose that minimizes the above cost function, the change should be in the opposite direction to the gradient. Indeed, the change in the pose is given as

$$\Delta P = -\lambda \frac{\partial E_h(P)^T}{\partial P} = -\lambda \Big(J_i^T(P) e_i(P) + J_p^T(P) e_p(P)\Big), \tag{18}$$

where in most practical situations we can set $\lambda_1 = \lambda_2 = \lambda$.

One can note that local minima may occurs if the gradient $\Big(J_i^T(P) e_i(P) + J_p^T(P) e_p(P)\Big) = 0$. This means that either $e_i(P) = e_p(P) = 0$, *i.e.* the global minima case, or (i) $e_i(P) \in Ker(J_i^T(P))$ and (ii) $e_p(P) \in Ker(J_p^T(P))$ in the same time. The case (ii) hold for global minima only where $Ker(J_p^T(P)) = \{0_{(6\times1)}\}$. In other words, problem relating to local image minima and Jacobian singularity will not affect the convergence here.

B. Gauss-Newton Method

To achieve the minimizing task defined by the hybrid cost function $E_h(P)$ given in (15), we substitute in (8) to get

$$\Delta P = -\lambda \mathcal{J}^{+}(P) \left[\begin{array}{c} e_i(P) \\ e_p(P) \end{array} \right], \tag{19}$$

where the matrix $\mathcal{J}(P)$ is given

$$\mathcal{J}(P) = \left[\begin{array}{c} \frac{\partial e_i(P)}{\partial P} \\ \frac{\partial e_p(P)}{\partial P} \end{array} \right] = \left[\begin{array}{c} J_i(P) \\ J_p(P) \end{array} \right]. \tag{20}$$

It can be shown that

$$\mathcal{J}^{+}(P) = \left(J_i^T J_i + I \right)^{-1} \left[J_i^T J_p^T \right]. \tag{21}$$

Where $Ker(J_p^T(P)) = \{0_{(6\times1)}\}$, the velocity vector will be nullified only at the desired position without suffering from any local minima.

To summarize, It is shown that minimizing the hybrid proposed function is realizable. This function $E_h(P)$ has a common global minima with each of $E_{2D}(P)$ and $E_{3D}(P)$. The minimization of $E_h(P)$ is done using NLSM owing to the non-linear part arising from the 2D image space. However, global minima is the optimal solution that can be asymptotically obtained after a limited number of iterations.

The integration between 2D and 3D can be biased using the scalar factors λ_1 and λ_2. This gives a sense in designing a high level rule that computes these factors λ_1 and λ_2. A proper computation of the integration factors can be used to improve the performance of the whole process by individually taking advantages of the 2D or the 3D features used in the error function. For example, it is possible to put $\lambda_1 = \frac{\omega_{2D}}{\omega_{2D}+\omega_{3D}}$ and $\lambda_2 = \frac{\omega_{3D}}{\omega_{2D}+\omega_{3D}}$. The weights λ_1 and λ_2 are the importance factors of the 2D and 3D informations respectively [12].

V. Results and Discussion

A. Application to Visual Servoing

The minimization methods presented in section III are widely used in the vision-based control, namely in visual servoing. Here, we will call the control law based on (18) as the Jacobian Transpose Control (JTC) method, while the control law based on method given in (19) will be called the Jacobian Pseudo-inverse Control (JPC) [10]. We present the simulation experiments where the proposed methods are compared to previous method like IBVS, PBVS, and hybrid methods, namely 21/2D visual servoing. These methods in addition to our proposed method are implemented in a simulation framework.

The comparison is carried out for a positioning task. The task is a general positioning task that contains a rotational and translational errors. This task is useful to evaluate the camera trajectory in the Cartesian space. The camera trajectories using our two JTC and JPC methods and methods like IBVS, PBVS and the hybrid 21/2D VS are compared here.

The servoing target object consists of six non-planar points. The object pose, and initial and desired camera pose with respect to a world reference frame are given in Tables II. The object point coordinates are given in Tables III. We assume a perspective camera model with unit aspect ratio of pixels.

	T_x	T_y	T_z		R	P	Y
Initial pose	2.15	-2.1	0.0		0.8	-0.6	-1.59
Desired pose	0.0	-1.5	0.0		0.0	0.0	-1.57
Object pose	0.0	0.0	0.0		0.0	0.0	-1.57

TABLE II

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

	X	Y	Z		X	Y	Z
P_1	0.0	0.0	0.0	P_2	0.25	0.25	0.0
P_3	-0.25	0.25	0.0	P_4	-0.25	-0.25	0.0
P_5	-0.25	0.25	-0.2	P_6	-0.2	-0.2	0.1

TABLE III

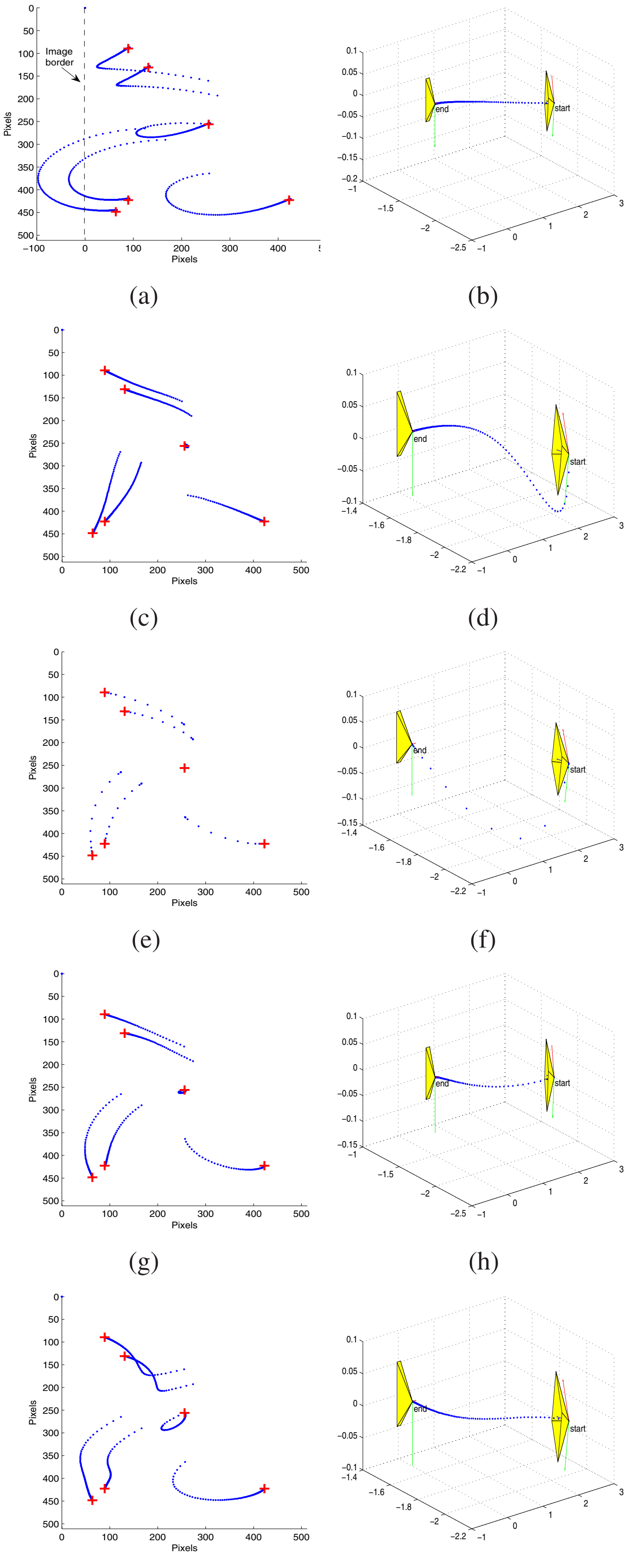
THE 3D POINTS COORDINATES IN THE OBJECT FRAME.

1) Camera trajectory in the Cartesian Space: Here, we consider the general task with both translational and rotational motion. First we present the results from PBVS. This method minimizes a function of only 3D visual information *i.e.* the pose 6-vector. The camera trajectory in the Cartesian space is a straight line. This is the shortest camera path. Figs 2(a) and 2(b) show the features trajectory in the image space and camera trajectory in the Cartesian space respectively. However the image trajectory is a complex curve and some features have got out of the camera field of view. In practice, this considered as process failure.

In contrast, IBVS that minimizes a function of only 2D visual information *i.e.* the 2D image coordinates of the feature, produces, in Figs 2(c) and 2(d) a very satisfactory image trajectory. This straight line image trajectory will strictly keep the object features visible during the servoing process. The camera trajectory is unpredictable and this may cause the arm of the robot to get out of its workspace.

The hybrid method used in this simulation *i.e.* 21/2D visual servoing, shows in Figs 2(e) and 2(f), an intermediate results. The image trajectory has been improved comparing to PBVS. Unfortunately, there is no much improvement in the Cartesian camera path, it is not straight line. Using PJC method produces a camera trajectory that is very much near to the straight line. See Fig 2(g). Similar results for the image trajectory are obtained as shown in Fig 2(f). There is no significant deviation from the straight line trajectory. Therefore, it is less probable to leave the camera field of view. TJC method produces a more complicated image trajectory but features are still alwayes visible (see Figs 2(i)). The camera path is approximatly straight line as shown in Fig 2(j).

The proposed method, which minimizes a hybrid (2D-3D) error function that span all the degrees of freedom of the Cartesian space, shows an improved performance in image space and Cartesian space together. In both minimization methods, gradient decent and Gauss-Newton, image features are less probable to leave the image while the camera performs less retreat in the Cartesian space. An improvement in the



(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j)

Fig. 2. Feature trajectories in the image space and the camera trajectory in the Cartesian space. PBVS in (a) and (b), IBVS in (c) and (d),21/2D visual servoing in (e) and (f), JPC visual servoing in (g) and (h), JTC visual servoing in (i) and (j). The desired positions of the image features are marked by +.

features behaviour in the image space for all considered points. Most of previous hybrid methods improve the trajectory of single point only from the image.

VI. Conclusion

We propose a hybrid cost function for solving computer vision problems from areas like augmented reality and Vision-based control. Solutions to these problems are based on linear and non-linear optimaization techniques. The proposed function is a weighted summation of two individual functions. One of them formulated in the image (2D) space and the other one in the Cartesian (3D) space. Experiments show considerable improvements in the bevaviour of the minimization process in both the image space and Cartesian space.

References

[1] E. Malis, F. Chaumette, and S. Boudet, “2 1/2 d visual servoing,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, April 1999.

[2] E. Malis and F. Chaumette, “Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods,” *IEEE Trans. on Robotics and Automation*, vol. 18, no. 2, pp. 176–186, April 2002.

[3] A. Comport, E. Marchand, M. Pressigout, and F. Chaumette, “Real-time markerless tracking for augmented reality: the virtual visual servoing framework,” *IEEE Trans. on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 615–628, July 2006.

[4] E. Marchand and F. Chaumette, “Virtual visual servoing: a framework for real-time augmented reality,” in *EUROGRAPHICS 2002 Conference Proceeding*, ser. Computer Graphics Forum, G. Drettakis and H.-P. Seidel, Eds., vol. 21(3), Saarebrcken, Germany, September 2002, pp. 289–298.

[5] D. Dementhon and L. Davis, “Model-based object pose in 25 lines of codes,” *International Journal of Computer Vision*, vol. 15, pp. 123–141, 1995.

[6] G. Malandain, S. Fernandez-Vidal, and J.-M. Rocchisani, “Physically based rigid registration of 3d free-form objects: Application to medical imaging,” INRIA, Sophi-Antipolis, France, Tech. Rep. INRIA/RR-2453-FR+ENG, 19951.

[7] L. Brown, “A survey if image registration tecniques,” *ACM Computing Surveys*, vol. 24, no. 1, pp. 325–376, December 1992.

[8] K. Hashimoto and H. Kimura, “L q optimal and nonlinear approaches to visual servoing,” in *Visual Servoing*, ser. World Scientific Series in Robotics and Automation Systems. World Scientific Press, 1993, vol. 7, pp. 165–198.

[9] E. Malis, “Hybrid vision-based robot control robust to large calibration errors on both intrinsic and extrinsic camera parameters,” in *European Control Conference, ECC’01*, Porto, Portugal, September 2001, pp. 2898–2903.

[10] ——, “Improving vision-based control using efficient second-order minimization techniques,” in *IEEE Int. Conf. on Robotics and Automation, ICRA’04*, New Orleans, USA, April 2004.

[11] B. Espiau, F. Chaumette, and P. Rives, “A new approach to visual servoing in robotics,” *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, July 1992.

[12] A. H. Abdul Hafez and C. V. Jawahar, “Probabilistic integration framework for improved visual servoing in image and cartesian spaces,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS’06*, Bejing, China, October 2006.