Neural Net Robotics Visual Servo: Learning The Epipolar Geometry

K. N. Al Muteb¹, E. A. Mattar², M. Al-Sulaiman³, H. Ramdane⁴, and M. Emaduddin⁵
¹ Comp & Information Sciences College, King Saud University, P. O. Box 51178, KSA
²College of Engineering, University of Bahrain, P. O. Box 13184, Kingdom of Bahrain
³,⁴,⁵ Comp & Information Sciences College, King Saud University, P. O. Box 51178, KSA

Abstract - Studies have shown that computations of visual kinematics relations are highly complicated and do require a considerable amount of time. This is due to dependent on Jacobians and massive inter-related relations. This hinders even complicated visual servo algorithm for real-time applications. In this respect, the proposed methodology is based on approximating the highly nonlinear relations and epipolar geometry relating changes of object visual features to changes in joint space of a robotics arm system, which are usually expressed in terms of kinematics relations, in addition to time-dependent Jacobian matrix. Artificial Neural system have been proposed for that purpose. A supervised learning artificial neutral network have been employed for learning the visual nonlinear kinematics relations. For validation, the concept have been applied to the well known Rives visual servo algorithm [1], with Two Scenes Epipolar Geometry. Results have shown that, highly accurate visual serving was achieved, while considerable amount of time has been reduced with the proposed methodology.

Keywords: Mobile Visual Navigation, Epipolar Geometry.

1 Introduction

Visual servoing aims to control a robotics system through artificial vision in a way as to manipulate an environment, comparable to human actions. Intelligence-based visual control (e.g. neural or fuzzy systems) has also been introduced by research community as a way to supply robotics system even with more cognitive capabilities, [2]. There have been few number of research on the field of intelligent visual robotics arm control. For instant, an Image Based Visual Servoing using Takagi-Sugeno fuzzy neural network controller has been proposed by Miao et. al. [3]. In this paper, a Takagi-Sugeno Fuzzy Neural Network Controller (TSFNNC) based Image Based Visual Servoing (IBVS) method is proposed. Firstly, the eigenspace based image compression method is explored, which is chosen as the global feature transformation method. After that, the inner structure, performance and training method of T-S neural network controller are discussed respectively. Besides, the whole architecture of the TS-FNNC is investigated. Panwar and Sukavanam in [4] have introduced Neural Network Based Controller for Visual Servoing of Robotic Hand Eye System. In [5], Gilles et. al. have proposed neural networks organizations to learn complex robotic functions. The study considers a general problem of function estimation with a modular approach of neural computing. In [6] an approximate for exploring motion control for eye-in-hand visual servoing was presented by Mariko and Masaaki. Their study proposes a visual servo control method for exploring motion of eye-in-hand robot to recognize a three-dimensional object. In [7] an adaptive visual servo regulation control for camera-in-hand configuration with a fixed camera extension was presented by Chen, et. al. In this paper, image-based regulation control of a robot manipulator with an uncalibrated vision system is discussed. To compensate for the unknown camera calibration parameters, a novel prediction error formulation is presented. To achieve the control objectives, a Lyapunov-based adaptive control strategy is employed.

Fig. (1) : Camera image frame and Epipolar Geometry for the PowerBot AGV.
The control development for the camera-in-hand problem is presented in detail and a fixed-camera problem is included as an extension. The EGT (Epipolar Geometry Toolbox) [8], was also created to grant MATLAB users with a broaden outline for a creation and visualization of multi-camera scenarios. Image Based Visual Servoing Using Takagi-Sugeno Fuzzy Neural Network Controller has been proposed by Miao et al. [9]. In their study, a T-S fuzzy neural controller based IBVS method was proposed. Eigenspace based image compression method is firstly explored which is chosen as the global feature transformation method. Inner structure, performance and training method of T-S neural network controller are discussed respectively. Besides that, the whole architecture of TS-FNNC is investigated. For robotics arm visual servo, this issue has been formulated as a function of object feature Jacobian. Feature Jacobian is a complicated matrix to compute for real-time applications. For more feature points in space, the issue of computing inverse of such matrix is even more hard to achieve.

2. Double Camera Scene Analysis

Epipolar Geometry

In this section, we shall consider an image resulting from two camera views. For two perspective views of the same scene taken from two separate viewpoints O₁ and O₂, this is illustrated in Fig.1. Also we shall assume that (m₁ and m₂) are representing two separate points in the views. In other words, perspective projection through O₁ and O₂, of the same point X₁, in both image planes λ₁ and λ₂. In addition, by letting (c₁) and (c₂) be the optical centers of two scene, the projection E₁ (E₂) of one camera center O₁ (O₂) onto the image plane of the other camera frame \( \lambda_1 (\lambda_2) \) is the epipole geometry. We can expressed such an epipole geometry in homogeneous coordinates in terms \( \tilde{E}_1 \) and \( \tilde{E}_2 \):

\[
\tilde{E}_1 = (E_{1x}, E_{1y}, 1)^T \\
\tilde{E}_2 = (E_{2x}, E_{2y}, 1)^T
\]

One of the main parameters of an epipolar geometry is the fundamental Matrix \( H \) (which is \( \Re \in 3 \times 3 \)). \( H \) conveys most of the information about the relative position and orientation \((t, R)\) between the two different views. Moreover, the fundamental matrix \( H \) algebraically relates corresponding points in the two images through the Epipolar Constraint. For instance, let the case of two views of the same 3-D point \( X_1 \), both characterized by their relative position and orientation \((t, R)\) and the internal, hence \( H \) is evaluated in terms of \( K_1 \) and \( K_2 \) (extrinsic camera parameters), [8] :

\[
H = K_2^{-T} [t], RK_1^{-1}
\]

In such a case, a 3-D point \( X_1 \) is projected onto two image planes, to points \( m_1 \) and \( m_2 \), as to constitute a conjugate pair. Given a point \( m_2 \) in left image plane, its conjugate point in the right image is constrained to lie on the epipolar line of \( m_1 \). The line is considered as the projection through \( C_2 \) of optical ray of \( m_1 \). All epipolar lines in one image plane pass through an epipole point. This is the projection of conjugate optical centre : \( \tilde{E}_1 = p_2 \left( \begin{array}{c} c_1 \\ 1 \end{array} \right) \) Parametric equation of epipolar line of \( \tilde{m}_2 \) gives \( \tilde{m}_2^T = \tilde{E}_2 + \lambda \cdot p_1, p_1^{-1} \tilde{m}_2 \). In image coordinates this can be expressed as :

\[
u = [m_2]_j = \left[ \begin{array}{c} \tilde{e}_j \\ \tilde{e}_j \end{array} \right] + \lambda \left[ \begin{array}{c} \tilde{v} \\ \tilde{v} \end{array} \right] \\
u = [m_1]_j = \left[ \begin{array}{c} \tilde{e}_j \\ \tilde{e}_j \end{array} \right] + \lambda \left[ \begin{array}{c} \tilde{v} \\ \tilde{v} \end{array} \right]
\]

here \( \tilde{v} = p_1, p_1^{-1} \tilde{m}_2 \) is a projection operator extracting the \( \lambda \) component from a vector. When \( (c_1) \) is in the focal plane of right camera, the right epipole is an infinity, and the epipolar lines form a bundle of parallel lines in the right image. Direction of each epipolar line is evaluated by derivative of parametric equations listed above with respect to \( (\lambda) \):

\[
\frac{du}{d\lambda} = \left[ \begin{array}{c} \tilde{e}_j \\ \tilde{e}_j \end{array} \right] - \lambda \left[ \begin{array}{c} \tilde{e}_j \\ \tilde{e}_j \end{array} \right] \\
\frac{dv}{d\lambda} = \left[ \begin{array}{c} \tilde{v} \\ \tilde{v} \end{array} \right] - \lambda \left[ \begin{array}{c} \tilde{v} \\ \tilde{v} \end{array} \right]
\]

The epipole is rejected to infinity once \( \left[ \tilde{E}_2 \right] = 0 \). In such a case, direction of the epipolar lines in right image doesn't
depend on any more. All epipolar lines becomes parallel to vector \( \begin{bmatrix} E_1 \end{bmatrix} \) \( \begin{bmatrix} E_2 \end{bmatrix} \). A very special occurrence is once both epipoles are at infinity. This happens once a line containing \((c_i, c_j)\), the baseline, is contained in both focal planes, or the retinal planes are parallel and horizontal in each image as in Fig. (1). The right pictures plot the epipolar lines corresponding to the point marked in the left pictures. This procedure is called rectification [8]. If cameras share the same focal plane the common retinal plane is constrained to be parallel to the baseline and epipolar lines are parallel.

3. Neural Net Based Image-Based Visual Servo Control (ANN-IBVS)

Over the last section we have focused more in single and double camera scenes, i.e. representing the robot arm visual sensory. In this section, we shall focus on Image-Based Visual Servo (IBVS) which uses locations of object features on image planes (epipolar) for direct visual feedback. For instant, re-consider Fig. 1, and Fig. 2, where it is desired to move a robotics arm in such away that camera’s view changes from (initial) to (final) view, and feature vector from \((\phi_0)\) to \((\phi_t)\). Here \((\phi_0)\) may comprise coordinates of vertices, or areas of the object to be tracked. Implicit in \((\phi_t)\) is the robot is normal to, and centered over features of an object, at a desired distance. Elements of the task are thus specified in image space. For a robotics system with an end-effector mounted camera, viewpoint and features are functions of relative pose of the camera to the target, \((\xi_t)\). Such function is usually nonlinear and cross-coupled. A motion of end-effectors DOF results in complex motion of many features. For instant, a camera rotation can cause features to translate horizontally and vertically on the same image plane, as related via the following relationship:

\[
\phi = f(\xi_t),
\]

Equ (7) is to be linearized. This is to be done around an operating point:

\[
\delta \phi = J(\xi_t) \delta \xi_t,
\]

\[
\dot{\xi}_t = J^T(\xi_t) \delta f
\]

In Equ (9), \( J(\xi_t) \) is the Jacobian matrix, relating rate of change in robot arm pose to rate of change in feature space. Various, this Jacobian is referred to as the feature Jacobian, image Jacobian, feature sensitivity matrix, or interaction matrix [10]. Assume that the Jacobian is square and non-singular, then:

\[
\dot{\xi}_t = J^T(\xi_t) \hat{f}_e
\]

will tend to move the robotics arm towards desired feature vector. In Equ (11), \( K^T \) is a diagonal gain matrix, and \( t \) indicates a time varying quantity. Object posture rates \( \dot{\xi}_t \) is converted to robot end-effector rates. A Jacobian, \( J(\xi_t) \) as derived from relative pose between the end-effector and camera, \((\xi_t)\) is used for that purpose. In this respect, a technique to determine a transformation between a robot’s end-effector and the camera frame is given by Tsai and Lenz [11]. In turn, an end-effector rates may be converted to manipulator joint rates using the manipulator’s Jacobian [12], as follows:

\[
\hat{\theta}_i = K^T J^T(\theta) \hat{J}_e \dot{\xi}_t
\]

\( \hat{\theta}_i \) represents the robot joint space rate. A complete closed loop equation can then be given by:

\[
\hat{\theta}_i = K^T J^T(\theta) \hat{J}_e \dot{\xi}_t (f_a - f(t))
\]

For achieving this task, an analytical expression of the error function is given by:

\[
\phi = Z^* \phi_0 + \gamma (I_x - Z^* Z) \frac{\partial \phi_0}{\partial X}
\]

Here, \( \gamma \in \mathcal{R}^+ \) and \( Z^* \) is pseudo inverse of the matrix \( Z \). \( Z \in \mathcal{R}^{m\times n} = \mathcal{R}(Z^T) = \mathcal{R}(J^T) \) and \( J \) is the Jacobian matrix of task function as \( J = \frac{\partial \phi}{\partial X} \). Due to modeling errors, such a closed-loop system is relatively robust in a possible presence of image distortions and kinematics parameter variations of the Puma 560 kinematics. A number of researchers also have demonstrated good results in using this image-based approach for visual servoing. It is always reported that, the significant problem is computing or estimating the feature Jacobian, where a variety of approaches have been used [12]. The proposed IBVS structure of Weiss [13,14], controls robot joint angles directly using measured image features. Nonlinearities include manipulator kinematics and dynamics as.

Fig. (3): Learning neural system
well as the perspective imaging model. Adaptive control was also proposed, since \( J^f(\theta) \), is pose dependent. In this study, changing relationship between robot pose, and image feature change is learned during the motion via a learning neural system. The learning neural system accepts a weighted set of inputs (stimulus) and responds. The four-layer feedforward neural network with \( n \) input units, \( m \) output units and \( N \) units in the hidden layer, is shown in the Fig. (3).

Fig. (3) exposes a one possible neural network architecture that has been used. In reference to the Fig. (3), every node is designed in such away to mimic its biological counterpart, the neuron. Interconnection of different neurons forms an entire grid of the used ANN that have the ability to learn and approximate the nonlinear visual kinematics relations. The used learning neural system composes of four layers. The input, output layers, and two hidden layers. If we denote \( ^w\mathbf{v}_c \) and \( ^w\mathbf{\omega}_c \) as the camera’s linear and angular velocities with respect to the robot frame respectively, motion of the image feature point as a function of the camera velocity is obtained by:

\[
\dot{\gamma} = \frac{\alpha_c}{p_x} \begin{bmatrix} 0 & 0 & ^wP_x & ^wP_y & ^wP_z & 0 \\ -1 & -1 & ^wP_x & ^wP_y & ^wP_z & ^wP_x \\ 0 & 0 & ^wR_x & 0 & 0 & ^w\mathbf{v}_c \\ 0 & 0 & 0 & ^wR_x & 0 & ^w\mathbf{\omega}_c \end{bmatrix}
\]

Instead of using coordinates \(^wP_x\) and \(^wP_z\) of the object feature described in camera coordinate frame, which are a priori unknown, it is usual to replace them by coordinates \((u)\) and \((v)\) of the projection of such a feature point onto the image frame.

### 4 Simulation: A Case Study

In this section the presented visual servoing is verified and discussed. The simulated system is presented in Fig. (2). During simulations the task has been performed using 6-DOF Puma manipulator with 6 revolute joints and a camera that can provide position information of the robot tip and the target in the robot workspace. The robot direct kinematics is given by the set of equations of Puma 560 robotics system, as documented in [15]. Kinematics and dynamics equations are already well known in the literature. For the purpose of comparison, the used example is based on visual servoing system developed by Rives [1]. The robotics system are has been servoing to follow an object that is moving in a 3-D working space. Object has been characterized by \((8\text{-features})\) marks, this has resulted in 24, \( R \in \mathbb{R}^{3\times3} \), size, feature Jacobian matrix.

INITIAL PHASE: FREE RUNNING SYSTEM, ANN DESIGN, AND LEARNING:

The foremost ambition of this visual servoing is to drive a 6-DOF robot arm, as simulated with Robot Toolbox [14], and equipped with a pin-hole camera, as simulated with Epipolar Geometry Toolbox, EGT [8], from a starting configuration toward a desired one using only image data provided during the robot motion. For the purpose of setting up the proposed method, Rives algorithm has been run a number of time before hand. In each case, the arm was servoing with different object posture and a desired location in the working space.

![Fig. (4): Top view: Actual object position and desired position.](image-url)
indicates some scene features. Fig. (5) shows the Robot arm-camera servoing and approaching towards a desired object posture. ANN was fed with defined pattern during arm movement. Epipolars have been used to evaluate visual features and the update during arm movement. Fig. (6) shows error between the Rives Algorithm and the proposed ANN based visual servo. Results suggest high accuracy of identical results, indicating that a learned neural system was able to servo the arm to desired posture. Difference in error was recorded within the range of $\left(2.5 \times 10^{-7}\right)$ for specific joint angles. Finally, Fig. (7) show migration of the eight visual features as seen over the camera image plan. Just the once the Puma robot arm was moving, concentration of features are located towards an end within camera image plane.

**5 Conclusions**

This article has been focused towards approximating complicated nonlinear kinematics and feature Jacobian matrices relating a robotics arm system movement in reference to an object displacement, as profoundly appear in closed loop visual servos systems. The presented approach avoids "heavily computed" kinematics relations. That was achieved whilst using a leaning artificial neural system, resulting in reduced computation time, usually a cumbersome for real-time applications. In addition, the proposed methodology depends on mergence of three MatLab-based Toolboxes together, the Robotics Toolbox, ANN toolbox, and the Epipolar Geometry Toolbox. While learning closed-loop visual and kinematics relations, a multi-layer neural net structure has been used for that purpose. The proposed methodology has proven to be an effective approach of approximation, and has reduced computational time usually needed to update visual feature Jacobian matrix as robotics system in motion.

**6 References**


ACKNOWLEDGMENT

This work is supported by NPST program by KING SAUD UNIVERSITY. Project: Number (08-ELE200-02). Kingdom of Saudi Arabia.