3

An Approach for Optimal Design of Robot Vision Systems

Kanglin Xu
New Mexico Tech
USA

1. Introduction

The accuracy of a robot manipulator's position in an application environment is dependent on the manufacturing accuracy and the control accuracy. Unfortunately, there always exist both manufacturing error and control error. Calibration is an approach to identifying the accurate geometry of the robot. In general, robots must be calibrated to improve their accuracy. A calibrated robot has a higher absolute positioning accuracy. However, calibration involves robot kinematic modeling, pose measurement, parameter identification and accuracy compensation. These calibrations are hard work and time consuming. For an active vision system, a robot device for controlling the motion of cameras based on visual information, the kinematic calibrations are even more difficult. As a result, even though calibration is fundamental, most existing active vision systems are not accurately calibrated (Shih et al., 1998). To address this problem, many researchers select self-calibration techniques. In this article, we apply a more active approach, that is, we reduce the kinematic errors at the design stage instead of at the calibration stage. Furthermore, we combine the model described in this article with a cost-tolerance model to implement an optimal design for active vision systems so that they can be used more widely in enterprise. We begin to build the model using the relation between two connecting joint coordinates defined by a DH homogeneous transformation.We then use the differential relationship between these two connecting joint coordinates to extend the model so that it relates the kinematic parameter errors of each link to the pose error of the last link. Given this model, we can implement an algorithm for estimating depth using stereo cameras, extending the model to handle an active stereo vision system. Based on these two models, we have developed a set of C++ class libraries. Using this set of libraries, we can estimate robot pose errors or depth estimation errors based on kinematic errors. Furthermore, we can apply these libraries to find the key factors that affect accuracy. As a result, more reasonable minimum tolerances or manufacturing requirements can be defined so that the manufacturing cost is reduced while retaining relatively high accuracy. Besides providing an approach to find the key factors and best settings of key parameters, we demonstrate how to use a cost-tolerance model to evaluate the settings. In this way, we can implement optimal *design for manufacturing*(DFM) in enterprises. Because our models are derived from the Denavit-Hartenberg transformation matrix, differential changes for the transformation matrix and link parameters, and the fundamental algorithm for estimating depth using stereo cameras, they are suitable for any manipulator or stereo active vision system. The remainder of this article is organized as follows. Section 2 derives the model for analyzing the effect of parameter errors on robot poses. Section 3 introduces the extended kinematic error model for an active vision system. It should be noted that this extended model is the main contribution of our article and that we integrate the robot differential kinematics into an active vision system. Section 4 provides more detailed steps describing how to use our approach. Section 5 discusses some issues related to the design of active vision systems and DFM. Section 6 presents a case study for a real active vision system and cost evaluation using a cost-tolerance model. Finally, Section 7 offers concluding remarks.

2. Kinematic Differential Model Derived from DH Transformation Matrix

A serial link manipulator consists of a sequence of links connected together by actuated joints (Paul, 1981). The kinematical relationship between any two successive actuated joints is defined by the DH (Denavit-Hartenberg) homogeneous transformation matrix. The DH homogeneous transformation matrix is dependent on the four link parameters, that is, θ_i, α_i, r_i, and d_i. For the generic robot forward kinematics, only one of these four parameters is variable. If joint i is rotational, the θ_i is the joint variable and d_i, α_i, and r_i are constants. If joint i is translational, the d_i is the joint variable and θ, α_i, and r_i are constants. Since there always exist errors for these four parameters, we also need a differential relationship between any two successive actuated joints. This relationship is defined by matrix $d\mathbf{A}_i$ which is dependent on $d\theta_i$, $d\alpha_i$, dr_i, and dd_i as well as θ_i, α_i, r_i, and d_i. Given the relationship between two successive joints $\mathbf{A}_i$ and differential relationship between two successive joints $d\mathbf{A}_i$, we can derive an equation to calculate the accurate position and orientation of the end-effector with respect to the world coordinate system for a manipulator with N degrees of freedom(N-DOF).
In this section, we will first derive the differential changes between two successive frames in subsection *2.1*. We then give the error model for a manipulator of N degrees of freedom with respect to the world coordinate system in subsection *2.2*.

2.1 The Error Relation between Two Frames

For an N-DOF manipulator described by the Denavit-Hartenberg definition, the homogeneous transformation matrix $\mathbf{A}_i$ which relates the (*i-1*)th joint to ith joint is (Paul, 1981)

$$\mathbf{A}_i = \begin{bmatrix} c\theta_i & -s\theta_i\ c\alpha_i & s\theta_i\ s\alpha_i & r_i c\theta_i \\ s\theta_i & c\theta_i\ c\alpha_i & -c\theta_i\ s\alpha_i & r_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where s and c refer to sine and cosine functions, and θ_i, α_i, r_i, and d_i are link parameters.
Given the individual transformation matrix $\mathbf{A}_i$, the end of an N-DOF manipulator can be represented as

$$\mathbf{T}_N = \mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_{N-1}\mathbf{A}_N \tag{2}$$

We will also use the following definitions. We define $\mathbf{U}_i = \mathbf{A}_i\mathbf{A}_{i+1}\cdots\mathbf{A}_N$ with $U_{N+1} = I$, and a homogeneous matrix

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{n}_i & \mathbf{o}_i & \mathbf{a}_i & \mathbf{p}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where $\mathbf{n}_i$, $\mathbf{o}_i$, $\mathbf{a}_i$ and $\mathbf{p}_i$ are 3×1 vectors.
Given the ith actual coordinate frame $\mathbf{A}_i$ and the ith nominal frame $\mathbf{A}_i^0$, we can obtain an additive differential transformation $d\mathbf{A}_i$

$$d\mathbf{A}_i = \mathbf{A}_i - \mathbf{A}_i^0. \tag{4}$$

If we represent the ith additive differential transformation $d\mathbf{A}_i$ as the ith differential transformation $\delta\mathbf{A}_i$ right multiplying the transformation $\mathbf{A}_i$, we can write

$$d\mathbf{A}_i = \mathbf{A}_i \, \delta\mathbf{A}_i. \tag{5}$$

In this case, the changes are with respect to coordinate frame $\mathbf{A}_i$.
Assuming the link parameters are continuous and differentiable we can represent $d\mathbf{A}_i$ in another way, that is

$$d\mathbf{A}_i = \frac{\partial \mathbf{A}_i}{\partial \theta_i} d\theta_i + \frac{\partial \mathbf{A}_i}{\partial \alpha_i} d\alpha_i + \frac{\partial \mathbf{A}_i}{\partial r_i} dr_i + \frac{\partial \mathbf{A}_i}{\partial d_i} dd_i. \tag{6}$$

Comparing (5) with (6), we obtain

$$\begin{aligned} \delta\mathbf{A}_i &= \mathbf{A}^{-1}\Big(\frac{\partial \mathbf{A}_i}{\partial \theta_i} d\theta_i + \frac{\partial \mathbf{A}_i}{\partial \alpha_i} d\alpha_i \\ &+ \frac{\partial \mathbf{A}_i}{\partial r_i} dr_i + \frac{\partial \mathbf{A}_i}{\partial d_i} dd_i\Big). \end{aligned} \tag{7}$$

For the homogeneous matrix, the inverse matrix of $\mathbf{A}_i$ is

$$\mathbf{A}_i^{-1} = \begin{bmatrix} \mathbf{n}_i^t & -\mathbf{p}_i \cdot \mathbf{n}_i \\ \mathbf{o}_i^t & -\mathbf{p}_i \cdot \mathbf{o}_i \\ \mathbf{a}_i^t & -\mathbf{p}_i \cdot \mathbf{a}_i \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{8}$$

By differentiating all the elements of equation (1) with respect to θ_i, α_i, r_i and d_i respectively, we obtain

$$\frac{\partial \mathbf{A}_i}{\partial \theta_i} = \begin{bmatrix} -s\theta_i & -c\theta_i\, c\alpha_i & c\theta_i\, s\alpha_i & -r_i\, s\theta_i \\ c\theta_i & -s\theta_i\, c\alpha_i & s\theta_i\, c\alpha_i & r_i\, c\theta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

$$\frac{\partial \mathbf{A}_i}{\partial \alpha_i} = \begin{bmatrix} 0 & s\theta_i\, s\alpha_i & s\theta_i\, c\alpha_i & 0 \\ 0 & -c\theta_i\, s\alpha_i & -c\theta_i\, c\alpha_i & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

$$\frac{\partial \mathbf{A}_i}{\partial r_i} = \begin{bmatrix} 0 & 0 & 0 & c\theta_i \\ 0 & 0 & 0 & s\theta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{11}$$

$$\frac{\partial \mathbf{A}_i}{\partial d_i} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{12}$$

Substituting equations (8), (9), (10), (11) and (12) into (7), we obtain

$$\delta\mathbf{A}_i = \begin{bmatrix} 0 & -c\alpha_i\, d\theta_i & s\alpha_i\, d\theta_i & dr_i \\ c\alpha_i\, d\theta_i & 0 & -d\alpha_i & u \\ -s\alpha_i\, d\theta_i & d\alpha_i & 0 & v \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{13}$$

where $u = r_i \; c\alpha_i \; d\theta_i + s\alpha_i \; dd_i$ and $v = r_i \; s\alpha_i \; d\theta_i + c\alpha_i \; dd_i$. Since $d\mathbf{A}_i = \mathbf{A}_i \; \delta\mathbf{A}_i$, therefore (Paul, 1981)

$$\begin{pmatrix} dx_i \\ dy_i \\ dz_i \end{pmatrix} = \begin{pmatrix} dd_i \\ d_i \; c\alpha \; d\theta_i + s\alpha_i \; dr_i \\ -d_i \; s\alpha \; d\theta_i + c\alpha_i \; dr_i \end{pmatrix} \tag{14}$$

$$\begin{pmatrix} \delta_{x_i} \\ \delta_{y_i} \\ \delta_{z_i} \end{pmatrix} = \begin{pmatrix} d\alpha_i \\ s\alpha \; d\theta_i \\ c\alpha \; d\theta_i \end{pmatrix} \tag{15}$$

where $\begin{pmatrix} dx_i & dy_i & dz_i \end{pmatrix}^t$ is the differential translation vector and $\begin{pmatrix} \delta_{x_i} & \delta_{y_i} & \delta_{z_i} \end{pmatrix}^t$ is the differential rotation vector with respect to frame $\mathbf{A}_i$.

Let $\mathbf{d}_i = (dx_i \;\; dy_i \;\; dz_i)^t$ and $\boldsymbol{\delta}_i = (\delta_{x_i} \;\; \delta_{y_i} \;\; \delta_{z_i})^t$. The differential vectors $\mathbf{d}_i$ and $\boldsymbol{\delta}_i$ can be represented as a linear combination of the parameter changes, which are

$$\mathbf{d}_i = \mathbf{k}_i^1 \; d\theta_i + \mathbf{k}_i^2 \; dd_i + \mathbf{k}_i^3 \; dr_i \tag{16}$$

and

$$\boldsymbol{\delta}_i = \mathbf{k}_i^2 \; d\theta_i + \mathbf{k}_i^3 \; d\alpha_i \tag{17}$$

where $\mathbf{k}_i^1 = (0 \;\; r_i c\alpha_i \;\; -r_i s\alpha_i)^t$, $\mathbf{k}_i^2 = (0 \;\; s\alpha_i \;\; c\alpha_i)^t$ and $\mathbf{k}_i^3 = (1 \; 0 \; 0)^t$.

2.2 Kinematic Differential Model of a Manipulator

Wu (Wu, 1984) has developed a kinematic error model of an N-DOF robot manipulator based on the differential changes $d\mathbf{A}_i$ and the error matrix $\delta\mathbf{A}_i$ due to four small kinematic errors at an individual joint coordinate frame.

Let $\mathbf{d}^N$ and $\boldsymbol{\delta}^N$ denote the three translation errors and rotation errors of the end of a manipulator respectively where $\mathbf{d}^N = (d\mathbf{x}^N \; d\mathbf{y}^N \; d\mathbf{z}^N)$ and $\boldsymbol{\delta}^N = (\boldsymbol{\delta}_x^N \; \boldsymbol{\delta}_y^N \; \boldsymbol{\delta}_z^N)$. From (Wu, 1984), we obtain

$$\begin{aligned} \begin{pmatrix} \mathbf{d}^N \\ \boldsymbol{\delta}^N \end{pmatrix} &= \begin{pmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \end{pmatrix} d\boldsymbol{\theta} + \begin{pmatrix} \mathbf{M}_2 \\ \mathbf{0} \end{pmatrix} d\mathbf{r} \\ &+ \begin{pmatrix} \mathbf{M}_3 \\ \mathbf{0} \end{pmatrix} d\mathbf{d} + \begin{pmatrix} \mathbf{M}_4 \\ \mathbf{M}_3 \end{pmatrix} d\boldsymbol{\alpha} \end{aligned} \tag{18}$$

where

$$\begin{aligned} d\boldsymbol{\theta} &= (d\theta_1 \quad d\theta_2 \quad \ldots \quad d\theta_N)^t \\ d\mathbf{r} &= (dr_1 \quad dr_2 \quad \ldots \quad dr_N)^t \\ d\mathbf{d} &= (dd_1 \quad dd_2 \quad \ldots \quad dd_N)^t \\ d\boldsymbol{\alpha} &= (d\alpha_1 \quad d\alpha_2 \quad \ldots \quad d\alpha_N)^t \end{aligned}$$

and $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_3$ and $\mathbf{M}_4$ are all $3 \times N$ matrices whose components are the function of N joint variables. The ith column of $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_3$ and $\mathbf{M}_4$ can be expressed as follows:

$$\mathbf{M}_1^i = \begin{bmatrix} \mathbf{n}_{n+1}^u \cdot \mathbf{k}_i^1 + (\mathbf{p}_{i+1}^u \times \mathbf{n}_{i+1}^u) \cdot \mathbf{k}_i^2 \\ \mathbf{o}_{i+1}^u \cdot \mathbf{k}_i^1 + (\mathbf{p}_{i+1}^u \times \mathbf{o}_{i+1}^u) \cdot \mathbf{k}_i^2 \\ \mathbf{a}_{i+1}^u \cdot \mathbf{k}_i^1 + (\mathbf{p}_{i+1}^u \times \mathbf{a}_{i+1}^u) \cdot \mathbf{k}_i^2 \end{bmatrix}$$

$$\mathbf{M}_2^i = \begin{bmatrix} \mathbf{n}_{n+1}^u \cdot \mathbf{k}_i^2 \\ \mathbf{o}_{i+1}^u \cdot \mathbf{k}_i^2 \\ \mathbf{a}_{i+1}^u \cdot \mathbf{k}_i^2 \end{bmatrix}$$

$$\mathbf{M}_3^i = \begin{bmatrix} \mathbf{n}_{n+1}^u \cdot \mathbf{k}_i^3 \\ \mathbf{o}_{i+1}^u \cdot \mathbf{k}_i^3 \\ \mathbf{a}_{i+1}^u \cdot \mathbf{k}_i^3 \end{bmatrix}$$

$$\mathbf{M}_4^i = \begin{bmatrix} (\mathbf{p}_{i+1}^u \times \mathbf{n}_{i+1}^u) \cdot \mathbf{k}_i^3 \\ (\mathbf{p}_{i+1}^u \times \mathbf{o}_{i+1}^u) \cdot \mathbf{k}_i^3 \\ (\mathbf{p}_{i+1}^u \times \mathbf{a}_{i+1}^u) \cdot \mathbf{k}_i^3 \end{bmatrix}$$

where $\mathbf{n}_{i+1}^u$, $\mathbf{o}_{i+1}^u$, $\mathbf{a}_{i+1}^u$ and $\mathbf{p}_{i+1}^u$ are four 3×1 vectors of matrix $\mathbf{U}_{i+1}$ which is defined as $\mathbf{U}_i = \mathbf{A}_i \mathbf{A}_{i+1} \cdots \mathbf{A}_N$ with $\mathbf{U}_{N+1} = I$.

Using the above equations, the manipulator's differential changes with respect to the base can be represented as

$$d\mathbf{T}_N = \begin{bmatrix} d\mathbf{n} & d\mathbf{o} & d\mathbf{a} & d\mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{19}$$

where

$$\begin{aligned} d\mathbf{n} &= \mathbf{o}_1^u\ \delta_z^N - \mathbf{a}_1^u\ \delta_y^N \\ d\mathbf{o} &= -\mathbf{n}_1^u\ \delta_z^N + \mathbf{a}_1^u\ \delta_x^N \\ d\mathbf{a} &= \mathbf{n}_1^u\ \delta_y^N - \mathbf{o}_1^u\ \delta_x^N \\ d\mathbf{p} &= \mathbf{n}_1^u\ d\mathbf{x}^N + \mathbf{o}_1^u\ d\mathbf{y}^N + \mathbf{a}_1^u\ d\mathbf{z}^N \end{aligned}$$

and $\mathbf{n}_1^u$, $\mathbf{o}_1^u$, $\mathbf{a}_1^u$ are four 3×1 vectors of matrix $\mathbf{U}_1$.

Finally, the real position and orientation at the end of the manipulator can be calculated by

$$\mathbf{T}_N^R = \mathbf{T}_N + d\mathbf{T}_N \tag{20}$$

where $\mathbf{T}_N = \mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_N$.

3. Extended Model for an Active Visual System

An active vision system, which has become an increasingly important research topic, is a robot device for controlling the motion of cameras based on visual information. The primary advantage of directed vision is its ability to use camera redirection to look at widely separated areas of interest at fairly high resolution instead of using a single sensor or array of cameras to cover the entire visual field with uniform resolution. It is able to interact with the environment actively by altering its viewpoint rather than observing it passively. Like an end effector, a camera can also be connected by a fixed homogeneous transformation to the last link. In addition, the structure and mechanism are similar to those of robots. Since an active visual system can kinematically be handled like a manipulator of N degrees of freedom, we can use the derived solutions in the last section directly.

In this section, we will first introduce the camera coordinate system corresponding to the standard coordinate system definition for the approaches used in the computer vision literature and describe a generic algorithm for location estimation with stereo cameras. We will then integrate them with the kinematic error model of a manipulator.

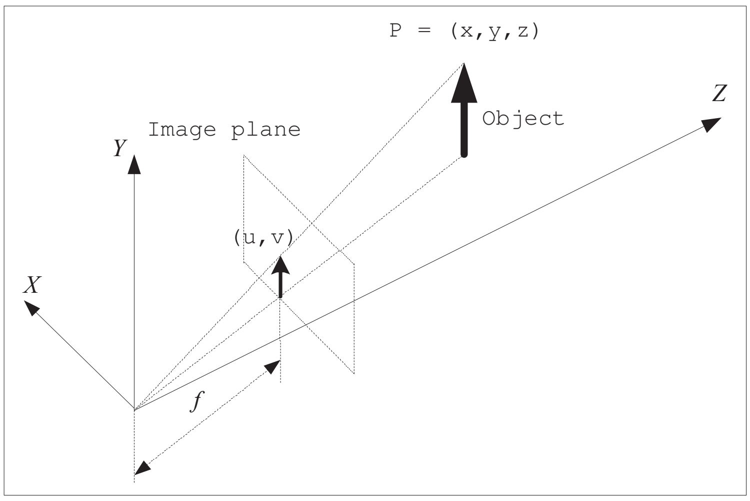


Fig. 1. The camera coordinate system whose x- and y-axes form a basis for the image plane, whose z-axis is perpendicular to the image plane (along the optical axis), and whose origin is located at distance f behind the image plane, where f is the focal length of the camera lens.

3.1 Pose Estimation with Stereo Cameras

We assign the camera coordinate system with x- and y-axes forming a basis for the image plane, the z-axis perpendicular to the image plane (along the optical axis), and with its origin located at distance f behind the image plane, where f is the focal length of the camera lens. This is illustrated in Fig. 1. A point, ${}^{c}\mathbf{p} = (x,\ y,\ z)^{t}$ whose coordinates are expressed with respect to the camera coordinate frame C, will project onto the image plane with coordinates $\mathbf{p}_i = (u,\ v)^t$ given by

$$\pi(x,\ y,\ z) = \begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{z} \begin{pmatrix} x \\ y \end{pmatrix} \tag{21}$$

If the coordinates of ${}^{x}\mathbf{p}$ are expressed relative to coordinate frame X, we must first perform the coordinate transformation

$${}^{c}\mathbf{p} = {}^{c}\mathbf{T}_{x}{}^{x}\mathbf{p}. \tag{22}$$

Let ${}^{a}\mathbf{T}_{c1}$ represent the pose of the first camera relative to the arbitrary reference coordinate frame $\mathbf{A}$. By inverting this transformation, we can obtain ${}^{c1}\mathbf{T}_{a}$, where

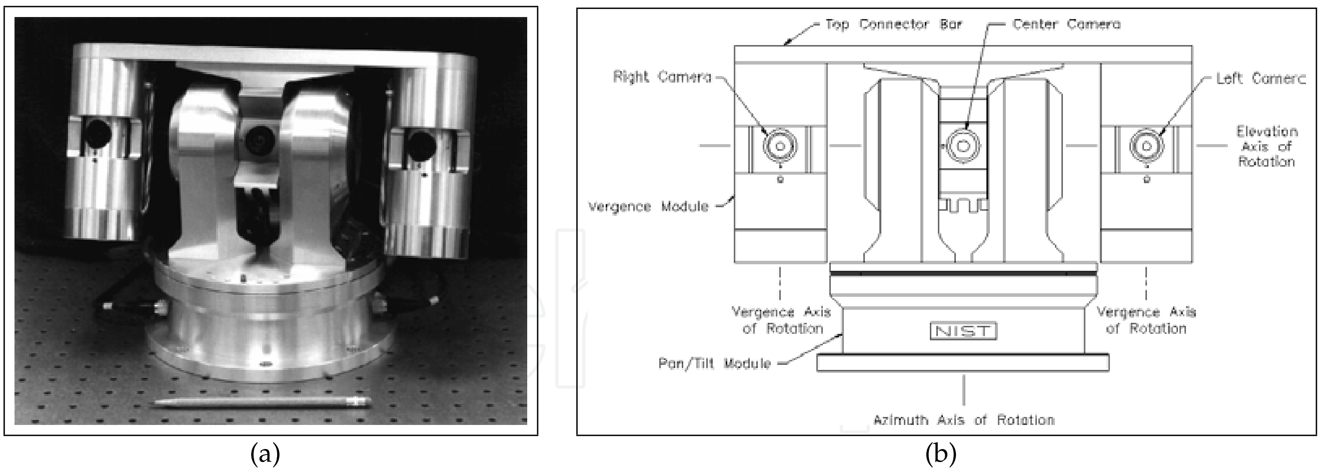
$${}^{c1}\mathbf{T}_{a} = \begin{bmatrix} {}^{c1}r_{11} & {}^{c1}r_{12} & {}^{c1}r_{13} & {}^{c1}t_{x} \\ {}^{c1}r_{21} & {}^{c1}r_{22} & {}^{c1}r_{23} & {}^{c1}t_{y} \\ {}^{c1}r_{31} & {}^{c1}r_{32} & {}^{c1}r_{33} & {}^{c1}t_{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

For convenience, let

$${}^{c1}\mathbf{R}_{1} = ({}^{c1}r_{11}\ {}^{c1}r_{12}\ {}^{c1}r_{13}) \tag{24}$$

$${}^{c1}\mathbf{R}_{2} = ({}^{c1}r_{21}\ {}^{c1}r_{22}\ {}^{c1}r_{23}) \tag{25}$$

$${}^{c1}\mathbf{R}_{1} = ({}^{c1}r_{31}\ {}^{c1}r_{32}\ {}^{c1}r_{33}) \tag{26}$$



(a) (b)

Fig. 2. *The TRICLOPS Active Vision System has four axes. They are pan axis, tilt axis, left vergence axis and right vergence axis. The pan axis can rotate around a vertical axis through the center of the base. The tilt axis can rotate around a horizontal line that intersects the base rotation axis. The left and right vergence axes intersect and are perpendicular to the tilt axis. These two pictures come from Wavering, Schneiderman, and Fiala (Wavering et al.). We would like to thank to the paper's authors.*

Given the image coordinates, Hutchinson, Hager and Corke (Hutchinson et al., 1996) have presented an approach to find the object location with respect to the frame **A** using the following equations:

$$\mathbf{A}_1 \cdot {}^{a}\mathbf{p} = \mathbf{b}_1 \tag{27}$$

where

$$\mathbf{A}_1 = \begin{pmatrix} f_1\, {}^{c1}\mathbf{R}_1 - u_1\, {}^{c1}\mathbf{R}_3 \\ f_1\, {}^{c1}\mathbf{R}_2 - v_1\, {}^{c1}\mathbf{R}_3 \end{pmatrix} \tag{28}$$

$$\mathbf{b}_1 = \begin{pmatrix} u_1\, {}^{c1}t_z - f_1\, {}^{c1}t_x \\ v_1\, {}^{c1}t_z - f_1\, {}^{c1}t_y \end{pmatrix} \tag{29}$$

Given a second camera at location ${}^{a}\mathbf{X}_{c2}$, we can compute ${}^{c2}\mathbf{X}_a$, $\mathbf{A}_2$ and ${}^{a}\mathbf{p}$ similarly. Finally we have an over-determined system for finding ${}^{a}\mathbf{p}$

$$\begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{pmatrix} {}^{a}\mathbf{p} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \tag{30}$$

where $\mathbf{A}_1$ and $\mathbf{b}_1$ are defined by (28) and (29) while $\mathbf{A}_2$ and $\mathbf{b}_2$ are defined as follows:

$$\mathbf{A}_2 = \begin{pmatrix} f_2\, {}^{c2}\mathbf{R}_1 - u_2\, {}^{c2}\mathbf{R}_3 \\ f_2\, {}^{c2}\mathbf{R}_2 - v_2\, {}^{c2}\mathbf{R}_3 \end{pmatrix} \tag{31}$$

$$\mathbf{b}_2 = \begin{pmatrix} u_2\, {}^{c2}t_z - f_2\, {}^{c2}t_x \\ v_2\, {}^{c2}t_z - f_2\, {}^{c2}t_y \end{pmatrix}. \tag{32}$$

3.2 Pose Estimation Based on an Active Vision System

As mentioned before, assuming that the camera frames are assigned as shown in Fig. 1 and that the projective geometry of the camera is modeled by perspective projection, a point ${}^c\mathbf{p} = ({}^c x\ {}^c y\ {}^c z)^t$, whose coordinates are expressed with respect to the camera coordinate frame will project onto the image plane with coordinates $(u\ v)^t$ given by

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{{}^c z} \begin{pmatrix} {}^c x \\ {}^c y \end{pmatrix} \tag{33}$$

For convenience, we suppose that f - the focal length of lens, does not have an error. This assumption is only for simplicity purposes and we will discuss this issue again in Section 5. Another problem is that the difference between the real pose of the camera and its nominal pose will affect the image coordinates. This problem is difficult to solve because the image coordinates are dependent on the depth of the object which is unknown. If we assume $f/{}^c z \ll 1$, we can regard them as high order error terms and ignore them. From these assumptions, we can obtain the real position and orientation of the left camera coordinate frame which is

$${}^a\mathbf{T}_{c1} = \mathbf{T}_1^R \mathbf{A}_{c1} \tag{34}$$

and those of the right camera coordinate frame which is

$${}^a\mathbf{T}_{c2} = \mathbf{T}_2^R \mathbf{A}_{c2} \tag{35}$$

In the above two equations, $\mathbf{T}_1^R$, $\mathbf{T}_2^R$ are the real poses of the end links and $\mathbf{A}_{c1}$, $\mathbf{A}_{c2}$ are two operators which relate the camera frames to their end links.

Given Equation (34) and (35), we can invert them to get ${}^{c1}\mathbf{T}_a$ and ${}^{c2}\mathbf{T}_a$. Then we can obtain an over-determined system using the method mentioned before. This system can be solved by the least squares approach as follows (Lawson & Hanson, 1995):

$${}^a\mathbf{p} = \left[(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \right] \mathbf{b} \tag{36}$$

where

$$\mathbf{A} = \begin{pmatrix} f_1\ {}^{c1}\mathbf{R}_1 - u_1\ {}^{c1}\mathbf{R}_3 \\ f_1\ {}^{c1}\mathbf{R}_2 - v_1\ {}^{c1}\mathbf{R}_3 \\ f_2\ {}^{c2}\mathbf{R}_1 - u_2\ {}^{c2}\mathbf{R}_3 \\ f_2\ {}^{c2}\mathbf{R}_2 - v_2\ {}^{c2}\mathbf{R}_3 \end{pmatrix} \tag{37}$$

$$\mathbf{b} = \begin{pmatrix} u_1\ {}^{c1}t_z - f_1\ {}^{c1}t_x \\ v_1\ {}^{c1}t_z - f_1\ {}^{c1}t_y \\ u_2\ {}^{c2}t_z - f_2\ {}^{c2}t_x \\ v_2\ {}^{c2}t_z - f_2\ {}^{c2}t_y \end{pmatrix} \tag{38}$$

If the superscript a in equations (34) and (35) indicates the world frame, we can calculate the position of $\mathbf{P}$ in world space.

4. Optimal Structure Design for Active Vision Systems

In the previous section, we derived an equation to calculate the real position when using a active vision with kinematic errors to estimate the location of a point P in the world space. Given this equation, it is straightforward to design the structure of an active vision system optimally. First, we can use $\mathbf{T}_1$ and $\mathbf{T}_2$ to replace $\mathbf{T}_1^R$ and $\mathbf{T}_2^R$ in equations (34) and (35). We then invert the resulting ${}^a\mathbf{T}_{c1}$ and ${}^a\mathbf{T}_{c2}$ to get ${}^{c1}\mathbf{T}_a$ and ${}^{c2}\mathbf{T}_a$. Finally we solve the over-determined system by the least squares approach to obtain the nominal pose of the cameras $\mathbf{P}^N$. The difference between these two results, *i.e.*

$$\mathbf{E} = \mathbf{P} - \mathbf{P}^N, \tag{39}$$

is the estimation error.

Note that the estimation errors are dependent on the joint variables and are a function of these joint variables. Consequently, a series of estimation errors can be obtained based on different poses of the stereo vision system. A curve describing the relationship between estimation errors and joint variables can be drawn. This curve can help us to analyze the estimation error or to design an active vision system. Inspired by the eyes of human beings and animals, we usually select a mechanical architecture with bilateral symmetry when we design a binocular or stereo active vision system. In this way, we can also simplify our design and manufacture procedures, and thus reduce the our design work and cost. We summarize our method described in the previous sections as follows:

1. Calculate transformation matrix $\mathbf{A}_i$ for each link based on the nominal size of the system and use them to calculate $\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_{N+1}$, where $\mathbf{U}_i = \mathbf{A}_i\mathbf{A}_{i+1}\cdots\mathbf{A}_N$, $\mathbf{U}_{N+1} = I$ and $\mathbf{T}_1 = \mathbf{U}_1$.
2. Calculate the operator $\mathbf{A}_{c1}$ which relates the camera frame to its end link and multiply it with $\mathbf{T}_1$ to obtain the nominal position and orientation of the camera coordinate frame $\mathbf{T}_{c1}$.
3. Repeat the above two steps to obtain the nominal position and orientation of the other camera coordinate frame $\mathbf{T}_{c2}$.
4. Invert frames $\mathbf{T}_{c1}$ and $\mathbf{T}_{c2}$ to obtain ${}^{c1}\mathbf{T}$ and ${}^{c2}\mathbf{T}$. Here we assume that $\mathbf{T}_{c1}$ represents the pose of the first camera relative to the world frame. Since they are homogeneous matrices, we can guarantee that their inverse matrices exist.
5. Derive Equation (36), an over-determined system using ${}^{c1}\mathbf{T}$ and ${}^{c2}\mathbf{T}$ and solve it by the least squares approach to find nominal location estimation $\mathbf{P}^N$.
6. Calculate four $3 \times N$ matrices $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_3$ and $\mathbf{M}_4$ in Equation (18), which are determined by the elements of matrices in $\mathbf{U}_i$ obtained in the first step. Since $\mathbf{U}_i$ is dependent on i^{th} link parameters θ_i, α_i, r_i and d_i, these four matrices are functions of the system link geometrical parameters as well as of the joint variable $\boldsymbol{\theta}$.
7. Based on performance requirements, machining capability and manufacturing costs, distribute tolerances to the four parameters of each link. Basically, the three geometrical tolerances $d\boldsymbol{\alpha}$, $d\mathbf{d}$, and $d\mathbf{r}$ affect manufacturing and assembling costs while the joint variable tolerance $d\boldsymbol{\theta}$ affects control cost.
8. Given four matrices $\mathbf{M_1}$, $\mathbf{M_2}$, $\mathbf{M_3}$, $\mathbf{M_4}$ and tolerances for each link, we can use Equation (19) to compute the total error with respect to the base frame. By adding it to the $\mathbf{T_1}$ obtained in the first step, we can have $\mathbf{T}_1^R$, the real position and orientation of the end link for the first camera. Similar to *Step 2*, we need to use the operator $\mathbf{A}_{c1}$ to do one more transformation to find the real position and orientation of camera coordinate frame $\mathbf{T}_{c1}^R$.

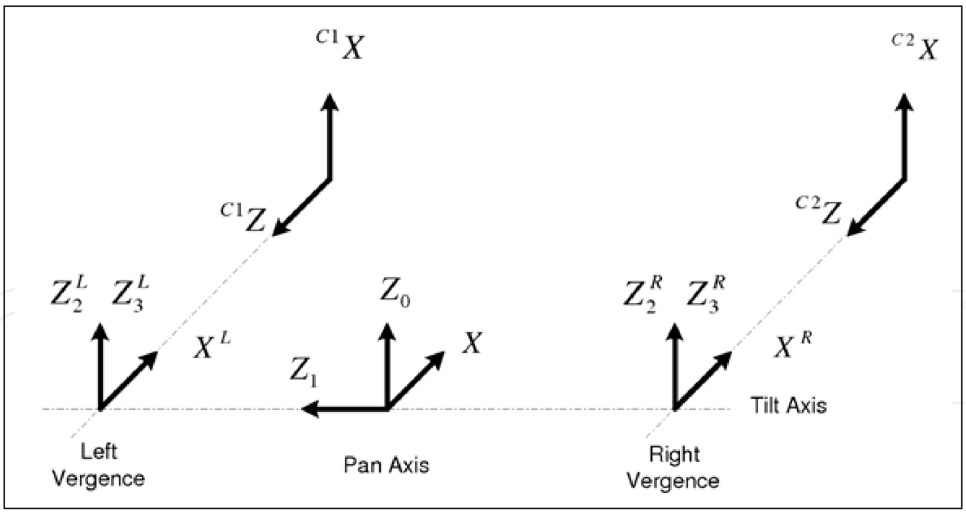


Fig. 3. Coordinate Frames for TRICLOPS - *Frame 0* for world frame; *Frame 1* for Pan; *Frame 2(left)* for left tilt; *Frame 2(right)* for right tilt; *Frame 3(left)* for left vergence; *Frame 3(right)* for right vergence. There are also two camera frames C_1 and C_2 whose original are located at distance f behind the image planes.

9. Repeating *Step 6*, *Step 7* and *Step 8* for the second camera, we can obtain $\mathbf{T}_{c2}^{R}$. As mentioned above, we usually have a symmetry structure for a stereo vision system. We assign the same tolerances $d\boldsymbol{\alpha}$, $d\boldsymbol{\theta}$, $d\mathbf{d}$ and $d\mathbf{r}$ for the two subsystems. Otherwise, the subsystem with low precision will dominate the whole performance of the system, and therefore we will waste the manufacturing cost for the subsystem with highest precision.
10. Similar to *Step 4* and *Step 5*, we can obtain the inverse matrices of $\mathbf{T}_{c1}^{R}$ and $\mathbf{T}_{c2}^{R}$, and use them to build another over-determined system. Solving it, we can have real pose $\mathbf{P}$ with respect to the world frame. The difference between $\mathbf{P}$ and $\mathbf{P}^N$ is the estimation error.
11. Repeat the above ten steps using another group of joint variables $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_N$ until we exhaust all the joint variables in the $\boldsymbol{\Theta}$ domain.
12. After exhausting all the joint variables in the $\boldsymbol{\Theta}$ domain, we have a maximum estimation error for assigning tolerances. Obviously, if this maximum does not satisfy the performance requirement, we need to adjust tolerances to improve the system precision and then go to *Step 1* to simulate the result again. On the other hand, if the estimation errors are much smaller than that required, we have some space to relax tolerance requirements. In this case, we also need to adjust tolerance in order to reduce the manufacturing cost and go to *Step 1* to start simulation. After some iterations, we can find an *optimal* solution.

Like many engineering designs, while it is true that we can learn from trial and error, those trials should be informed by something more than random chance, and should begin from a well thought out design. For example, we can initialize the tolerances based on the previous design experience or our knowledge in the manufacturing industry. The theory and model described in this article provide a tool for the design of such an active vision system.

Plan #	$d\theta$	$d\alpha$	$d\mathbf{d}$	$d\mathbf{r}$
1	$(.8^\circ\ .8^\circ\ .8^\circ)^T$	$(.8^\circ\ .8^\circ\ .8^\circ)^T$	$(.005\ .005\ .005)^T$	$(.005\ .005\ .005)^T$
2	$(.8^\circ\ .8^\circ\ .5^\circ)^T$	$(.8^\circ\ .8^\circ\ .5^\circ)^T$	$(.005\ .005\ .005)^T$	$(.005\ .005\ .005)^T$
3	$(.8^\circ\ .5^\circ\ .8^\circ)^T$	$(.8^\circ\ .5^\circ\ .8^\circ)^T$	$(.005\ .005\ .005)^T$	$(.005\ .005\ .005)^T$
4	$(1.1^\circ\ 1.1^\circ\ .5^\circ)^T$	$(1.1^\circ\ 1.1^\circ\ .5^\circ)^T$	$(.005\ .005\ .005)^T$	$(.005\ .005\ .005)^T$
5	$(.8^\circ\ .8^\circ\ .8^\circ)^T$	$(.8^\circ\ .8^\circ\ .8^\circ)^T$	$(.05\ .05\ .05)^T$	$(.05\ .05\ .05)^T$
6	$(.8^\circ\ .8^\circ\ .8^\circ)^T$	$(.8^\circ\ .8^\circ\ .8^\circ)^T$	$(.5\ .5\ .5)^T$	$(.5\ .5\ .5)^T$

Table 1. The Tolerances for Link Parameters (length unit: *in* and angle unit: *deg*)

Feature Category	A	k	δ_0	g_0	δ_a	δ_b
Ext. Rotational Sur	3.96	-22.05	0.0	0.79	0.0038	0.203
Hole	1.8	-20.08	-0.025	1.55	0.0038	0.203
Plane	1.73	-12.99	-0.025	0.79	0.0038	0.203
Location	0.68	-12.20	0.0	1.25	0.0038	0.203

Table 2. The parameters of Exponential Function A, k, δ_0, g_0, δ_a, δ_b (δ unit: mm) for four common manufacturing process (Dong & Soom, 1990)

5. Some Issues about Design of Active Vision Systems

As mentioned, the estimation errors are dependent on four link parameters $\boldsymbol{\theta}$, $\boldsymbol{\alpha}$, $\mathbf{r}$, $\mathbf{d}$ and their errors $d\boldsymbol{\theta}$, $d\boldsymbol{\alpha}$, $d\mathbf{r}$, $d\mathbf{d}$. Besides, they are also dependent on the focal length of the camera lens and its error. Note that although, for simplicity purposes, we do not include the errors of focal length in our model, adding them is straightforward. Since the cameras are located in the most end frames, we can obtain the effect of error of focal length directly by substituting the nominal focal length for the real one.

The four link parameters affect the structure and performance of active vision system. For example, the parameter $\boldsymbol{\theta}$ can determine the motion range of each link, and therefore affect the view space and domain of the vision system. These parameters should be given based on specification and performance requirements. The errors of link parameters affect not only pose estimation precision and performance but the manufacturing process, and therefore the manufacturing cost as well.

Basically, $d\boldsymbol{\alpha}$ is orientation tolerance while $d\mathbf{r}$ and $d\mathbf{d}$ are location tolerances. They affect manufacturing cost and assembly cost. The joint variable $d\boldsymbol{\theta}$ affects control cost. When we assign errors of link parameters, the bottom line is to satisfy the function requirement of the active system. In other words, "*...tolerances must be assigned to the component parts of the mechanism in such a manner that the probability that a mechanism will not function is zero...*"[1].

On the other hand, we must consider manufacturing cost when we design active vision systems since even for the same tolerances, different tolerance types can also affect manufacturing cost. For example, form tolerances are more difficult to machine and measure than size tolerances, and so are holes tolerances more difficult than shaft tolerances.

The final factor discussed in this article, which affects the systems precision and cost, is the focal length of camera. Since the differences between prices and performances for different cameras in the current market are big, selecting and purchasing a suitable camera is also a primary task for the system design. To do trade-off studies before the active system is built,

[1] Evans(1974)

we need a simulation tool for evaluating and optimizing our design. We need to use it to increase the understanding of how each error affects system performance and design the active vision system in terms of various parameters. Fortunately, the model in this article makes this simulation possible. Actually, we have developed a C++ class library to implement a simple tool. With it we can do experiments with various alternatives and obtain data indicating the best settings of key parameters.

6. TRICLOPS - A Case Study

In this section, we apply the model described above to a real active vision system - *TRICLOPS* as shown in Fig. 2[2]. First, we provide six design plans with tolerances assigned for all link parameters and analyze how the tolerances affect the pose estimation precision using our approach. We then compare the cost of each design plan based on an exponential cost-tolerance function. Please note that we do not give a complete design which is much more complicated than described here, and therefore beyond this article's range. We just want to demonstrate how to use our model to help to design active vision systems or analyze and estimate kinematic error.

TRICLOPS has four mechanical degrees of freedom. The four axes are: pan about a vertical axis through the center of the base, tilt about a horizontal line that intersects the base rotation axis and left and right vergence axes which intersect and are perpendicular to the tilt axis (Fiala et al., 1994). The system is configured with two $0.59(in)$ vergence lenses and the distance between the two vergence axes is $11(in)$. The ranges of motion are $\pm 96.3(deg)$ for the pan axis, from $+27.5(deg) to - 65.3(deg)$ for the tilt axis, and $\pm 44(deg)$ for the vergence axes. The image coordinates in this demonstration are arbitrarily selected as $u = -0.2$ and $v = 0.2$. The assigned link frames are shown in Fig. 3.

6.1 Tolerances vs. Pose Estimation Precise

As mentioned, the errors are dependent on the variable parameters. We let the three variables change simultaneously within their motion ranges, as shown in Fig 4. In this experiment, we have six design plans as shown in Table 1. The results corresponding to these six plans are shown in Fig. 5 in alphabetical order of sub-figures. If all the translational parameter errors are $0.005(in)$ and all angular parameter errors are $0.8(deg)$, from Fig. 5(a), we know that the maximum relative error is about 6.5%. Referring to Fig. 5(b), we can observe that by adjusting $d\theta_3$ and $d\alpha_3$ from 0.8(deg) to 0.5(deg), the maximum relative error is reduced from 6.5% to 5.3%. But adjusting the same amount for α_2 and θ_2, the maximum percentage can only reach 5.8%, as shown in Fig. 5(c). So the overall accuracy is more sensitive to α_3 and θ_3. As shown in Fig. 5(d), if we improve the manufacturing or control requirements for α_3 and θ_3 from $0.8(deg)$ to $0.5(deg)$ and at the same time reduce the requirements for α_1, α_2, θ_1 and θ_2 from 0.8(deg) to 1.1(deg), the overall manufacturing requirement is reduced by 0.6 (deg) while the maximum error is almost the same. From an optimal design view, these tolerances are more reasonable. From Fig. 5(e), we know that the overall accuracy is insensitive to translational error. From the design point of view, we can assign more translational tolerances to reduce the manufacturing cost while retaining relatively high accuracy.

[2] Thanks to Wavering, Schneiderman, and Fiala (Wavering et al.), we can present the TRICLOPS pictures in this article.

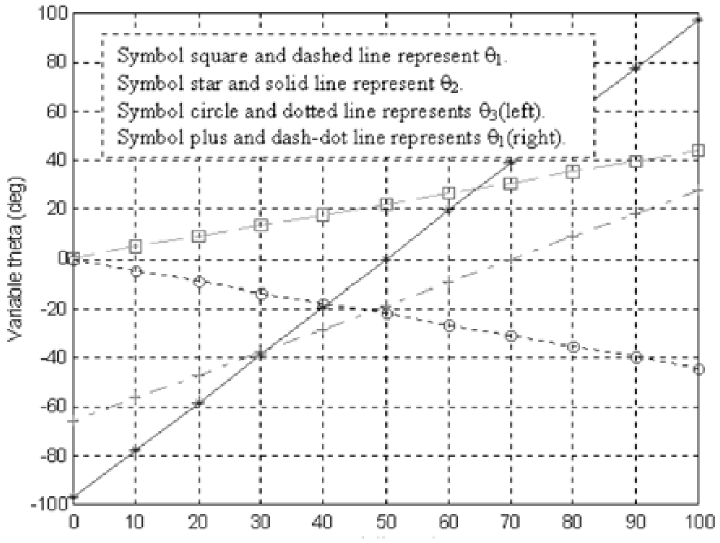


Fig. 4. Simulation Points - The pan axis whose range is from -96.3° to $+96.3^\circ$, tilt axis whose range is from -65.3° to $+27.5^\circ$, and two vergence axes whose ranges are from -44° to $+44^\circ$ rotate simultaneously.

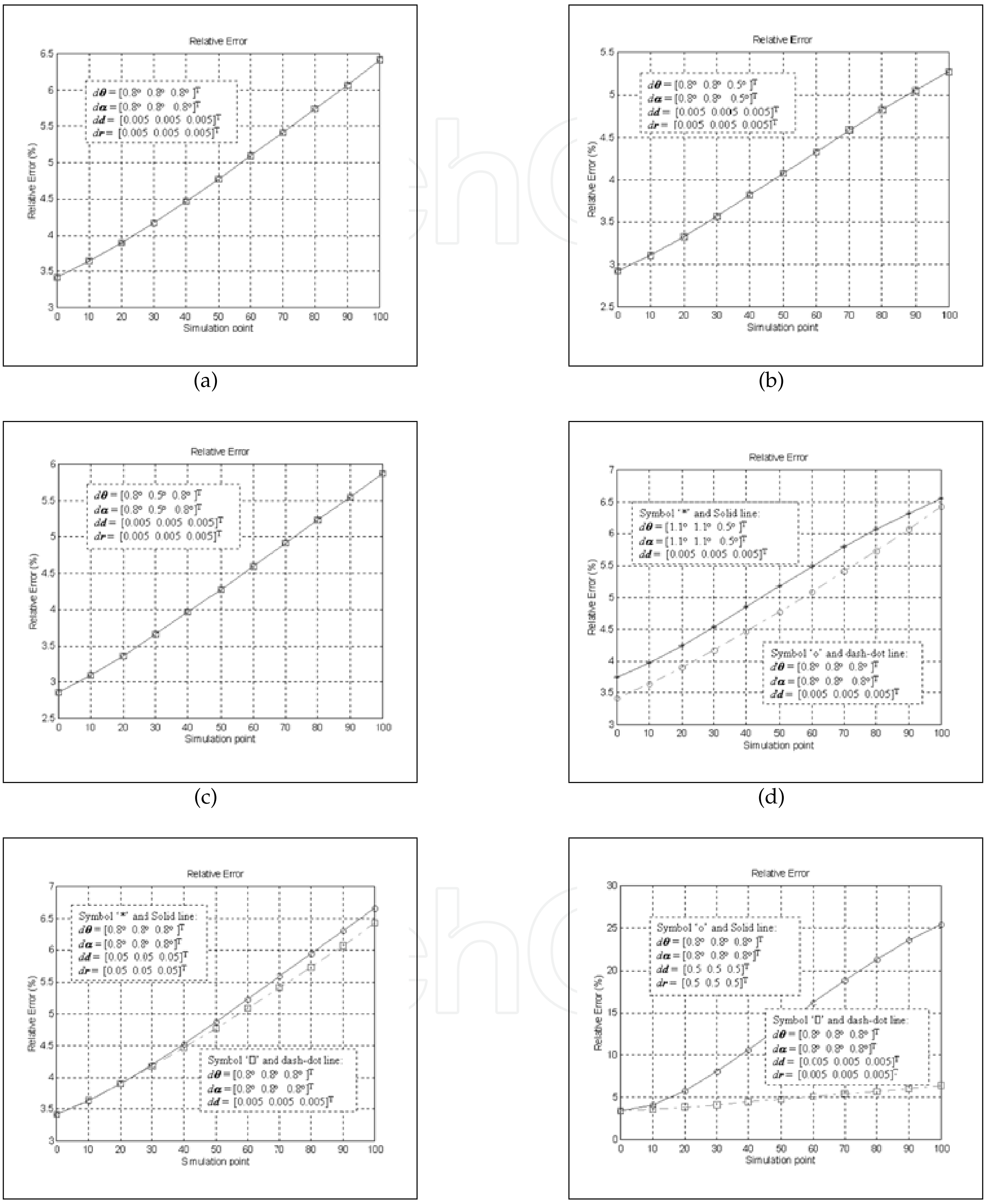
6.2 Tolerances vs. Manufacturing Cost

For a specific manufacturing process, there is a monotonic decreasing relationship between manufacturing cost and precision, called the *cost tolerance relation*, in a certain range. There are many cost tolerance relations, such as *Reciprocal Function*, *Sutherland Function*, *Exponential/Reciprocal Power Function*, *Reciprocal Square Function*, *Piecewise Linear Function*, and *Exponential Function*. Among them, the *Exponential Function* has proved to be relatively simple and accurate (Dong & Soom, 1990). In this section, we will use the *exponential function* to evaluate the manufacturing cost. The following is the mathematical representation of the exponential cost-tolerance function (Dong & Soom, 1990).

$$g(\delta) = Ae^{-k(\delta-\delta_0)} + g_0 \quad (\delta_0 \leq \delta_a < \delta < \delta_b) \tag{40}$$

In the above equation, A, δ_0, and g_0 determine the position of the cost-tolerance curve, while k controls the curvature of it. These parameters can be derived using a curve-fitting approach based on experimental data. δ_a and δ_b define the lower and upper bounds of the region, respectively, in which the tolerance is economically achievable. For different manufacturing process, these parameters are usually different. The parameters are based on empirical datum for four common feature categories *external rotational surface*, *hole*, *plane*, and *location*, shown in Table 2 are from (Dong & Soom, 1990) . For convenience, we use the average values of these parameters in our experiment. For angular tolerances, we first multiply them by unit length to transfer them to the length error, and then multiply the obtained cost by a factor 1.5 [3]. With these assumptions, we can obtain the relative total manufacturing costs, which are

[3] Angular tolerances are harder to machine, control and measure than length tolerances.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 5. *Experimental Results*

14.7, 14.9, 14.9, 14.5, 10.8 and 10.8 for the plans one through six mentioned above, respectively. Note that for *Plan 5* and *Plan 6* the length tolerances, after unit conversion, are greater than parameter δ_b, and therefore are beyond the range of *Exponential Function*. So we can ignore the *fine* machining cost since their tolerance may be achieved by rough machining such as forging. Compared with *Plan 1*, *Plan 2*, *Plan 3* and *Plan 4* do not change cost too much while *Plan 5* and *Plan 6* can decrease machining cost by 26%. From the analysis of the previous section and Fig. 5(e), we know that *Plan 5* increases system error a little while *Plan 6* is obviously beyond the performance requirement. Thus, *Plan 5* is a relatively optimal solution.

7. Conclusions

An active vision system is a robot device for controlling the optics and mechanical structure of cameras based on visual information to simplify the processing for computer vision. In this article, we present an approach for the optimal design of such active vision systems. We first build a model which relates the four kinematic errors of a manipulator to the final pose of this manipulator. We then extend this model so that it can be used to estimate visual feature errors. This model is generic, and therefore suitable for analysis of most active vision systems since it is directly derived from the DH transformation matrix and the fundamental algorithm for estimating depth using stereo cameras. Based on this model, we developed a standard C++ class library which can be used as a tool to analyze the effect of kinematic errors on the pose of a manipulator or on visual feature estimation. The idea we present here can also be applied to the optimized design of a manipulator or an active vision system. For example, we can use this method to find the key factors which have the most effect on accuracy at the design stage, and then give more suitable settings of key parameters. We should consider assigning high manufacturing tolerances to them because the accuracy is more sensitive to these factors. On the other hand, we can assign low manufacturing tolerances to the insensitive factors to reduce manufacturing cost. In addition, with the help of a cost-tolerance model, we can implement our *Design for Manufacturing* for active vision systems. We also demonstrate how to use this software model to analyze a real system *TRICLOPS*, which is a significant proof of concept. Future work includes a further analysis of the cost model so that it can account for control errors.

8. Acknowledgments

Support for this project was provided by DOE Grant *#DE-FG04-95EW55151*, issued to the UNM Manufacturing Engineering Program. Figure 2 comes from (Wavering et al., 1995). Finally, we thank Professor Ron Lumia of the Mechanical Engineering Department of the University of New Mexico for his support.

9. References

Dong, Z. & Soom, A. (1990). Automatic Optimal Tolerance Design for Related Dimension Chains. *Manufacturing Review*, Vol. 3, No.4, December 1990, 262-271.

Fiala, J.; Lumia, R.; Roberts, K.; Wavering, A. (1994). TRICLOPS: A Tool for Studying Active Vision. *International Journal of Computer Vision*, Vol 12, #2/3, 1994.

Hutchinson, S.; Hager, G.; Corke, P. (1996). A Tutorial on Visual Servo Control. *IEEE Trans. On Robotics and Automation*, Vol. 12, No.5, Oct. 1996, 651-670.

Lawson, C. & Hanson, R. (1995). *Solving Least Squares Problems*, SIAM, 1995.

Mahamud, S.; Williams, L.; Thornber, K.; Xu, K. (2003). Segmentation of Multiple Salient Closed Contours from Real Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 4, April 2003.

Nelson, B. & Khosla, P. (1996). Force and Vision Resolvability for Assimilating Disparate Sensory Feedback. *IEEE Trans. on Robotics and Automation*, Vol 12, No. 5, October 1996, 714-731.

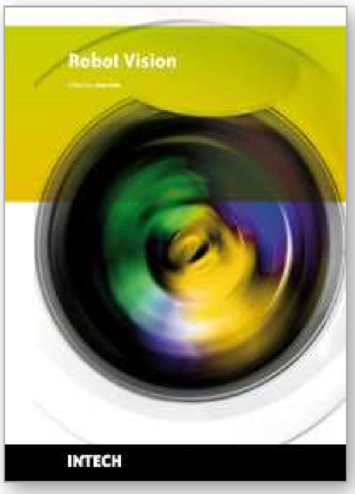
Paul, R. (1981) *Robot Manipulators: Mathematics, Programming, and Control*, Cambridge, Mass. MIT Press, 1981.

Shih, S.; Hung, Y.; Lin, W. (1998). Calibration of an Active Binocular Head. *IEEE Trans. On Systems, Man , and Cybernetics - part A: Systems and Humans*, Vol 28, No.4, July 1998, 426-442.

Wavering, A.; Schneiderman, H.; Fiala, J. (1995). High-Performance Tracking with TRICLOPS. *Proc. Second Asian Conference on Computer Vision, ACCV'95*, Singapore, December 5-8, 1995.

Wu, C. (1984). A Kinematic CAD Tool for the Design and Control of a Robot Manipulator, *Int. J. Robotics Research*, Vol. 3, No.1, 1984, 58-67.

Zhuang, H. & Roth, Z. (1996). *Camera-Aided Robot Calibration*, CRC Press, Inc. 1996.



Robot Vision
Edited by Ales Ude

ISBN 978-953-307-077-3
Hard cover, 614 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

The purpose of robot vision is to enable robots to perceive the external world in order to perform a large range of tasks such as navigation, visual servoing for object tracking and manipulation, object recognition and categorization, surveillance, and higher-level decision-making. Among different perceptual modalities, vision is arguably the most important one. It is therefore an essential building block of a cognitive robot. This book presents a snapshot of the wide variety of work in robot vision that is currently going on in different parts of the world.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821