Proceedings of the 2011 IEEE
International Conference on Mechatronics and Automation
August 7 - 10, Beijing, China

A Bilinear Model Based Solution to Object Pose Estimation with Monocular Vision for Grasping

Zhicai Ou
Institute of Automation
Chinese Academy of Sciences
Beijing, 100190, China
zhicai.ou@ia.ac.cn

Wei Liu and Jianhua Su
Institute of Automation
Chinese Academy of Sciences
Beijing, 100190, China
{wei.liu & jianhua.su}@ia.ac.cn

Abstract – **Object grasping is an important step in robotic applications for subsequent operations, such as delivery and assembly. Automatic object pose estimation with monocular vision provides useful visual cues for grasping and makes it flexible. However, some of the pose factors, such as the pitch angle and the yaw angle, are difficult to estimate from the monocular vision. In this paper, a modified bilinear model [9] is used to separate the pitch factor and the yaw factor from the object image so as to estimate the particular pitch angle and yaw angle. The iterative singular vector decomposition (SVD) in bilinear model fitting imposes a great computation burden. Thus, a random projection algorithm [17] is used to reduce the dimension of the data while preserving the performance of the bilinear model. A weighted Euclidian distance based factor identification method, which discriminates the importance of the elements of the factor parameters, is presented to improve the robustness of the factor identification. Furthermore, with the pitch angle and the yaw angle estimated from the modified bilinear model, a three-step object pose estimation solution is proposed. Experiments are performed to verify the proposed pose estimation solution.**

Index Terms* – *bilinear model, pose estimation, monocular vision, grasping

I. Introduction

Object grasping plays an essential role in robotic applications for subsequent operations, such as delivery and assembly. However, many robotic grasping tasks are limited to restricted environments. For instance, work-pieces in assembly lines are usually placed at a fixed position and orientation for robots to grasp. Whereas, machine vision systems which can detect and locate objects robustly make the grasping tasks flexible by allowing the objects to be placed at arbitrary positions.

To estimate the pose of an object placed on the assembly line, lots of algorithms have been proposed. One category of the popular methods is based on active illuminations which project active pattern on the object to assist robust feature extraction. Early works on pose estimation by using active illuminations are review by Batlle et al. [1]. In recent years, Scharstein and Szeliski [2] presented a method to acquire high-complexity stereo image pairs with highly precise correspondence information and estimate the depth of the field precisely. Breitenreicher and Schnörr [3] presented a two-stage method for detecting and registering multiple industrial 3D objects in unstructured range images. Besides the structured light, the multi-flash camera provides a new way for pose estimation of the object [4]. One of the advantages of these methods is high-accuracy in pose estimation.

Another popular category of the methods for object pose estimation is based on monocular vision systems. Compared with the active illumination based systems, monocular vision systems have the advantages of simple implementation, low-cost and movable to cover large space. Haralick et al. addressed solutions to four different pose estimation problems for monocular visions [5]. Dementhon and Davis [6] presented a method to estimate the object pose from a single image by using four or more noncoplanar feature points of the object and their relative geometry on the object. Ansar and Daniilidis [7] gave a general framework to estimate the poses of a limited number of n points and n lines simultaneously. These methods utilize feature points, object contours and some geometric information of the object to estimate the pose. Murase and Nayar [8] gave an appearance based method for 3D object pose estimation in different illuminations. Appearance base methods are more robust to noises in the image. But it is affected by a number of coupled factors, such as object shape, object pose, illumination conditions and reflection properties. This makes the pose estimation difficult and inspires us to separate interested factors from the appearance.

Thus, the bilinear model [9], which is first proposed by Tenenbaum and Freeman to separate the *independent* “content” and “style” factors of the observations in perceptual systems, is used in this work. After being presented, the bilinear model has been utilized to solve different problems in many fields of research, for instance, facial recognition [10, 11], 3D facial expression recognition [12], facial expression synthesis [13], sparse coding of location and content in natural images [14], gait recognition [15], speaker adaption [16], and so on. But, it is the first time to use the bilinear model for pose estimation.

In our work, the pitch and the yaw angle are considered as the independent “style factor” and “content factor”, respectively, and are separated using a bilinear model. The bilinear model is constructed using a SVD based iterative algorithm offline. And the pitch angle and the yaw angle are estimated by comparing the computed factor parameters with the fitted factor parameters. In order to overcome the dimension problem in model fitting, a random projection [17] is applied before model construction. Although some

978-1-4244-8115-6/11/$26.00 ©2011 IEEE

information is lost in the dimensionality reduction process, experiments show that the random projection keeps the efficiency of the bilinear model. The elements of the fitted parameter vectors are stacked in a descending order according to their contributions to the model. Thus, a modified factor identification method emphasizing the important elements is presented.

With the bilinear model estimating the pitch and the yaw angle, some machine vision techniques are used to estimate the other parameters of the pose. Combining the techniques mentioned above, a three-step solution to estimate the pose of the object is proposed in this paper.

The remainder of this work is organized as follows: the object grasping system with monocular vision is briefly introduced in section II. The proposed three-step object pose estimation solution is outlined in section III. In section IV, the construction of the bilinear model for pitch and yaw angle estimation is presented. Experiments are presented in section V. Section VI gives the conclusions.

II. Monocular Vision for Grasping

Monocular vision systems provide visual cues for locating the object lied on conveyor belt and allow the object to be placed at arbitrary position. Fig. 1 shows a monocular vision system for robot to grasp work-pieces placed on an assembly line. A CCD camera mounted on the robot captures the work-piece placed on the conveyor belt and estimate the pose of the work-piece. And then, the robot grasps the work-piece with the visual information. However, a number of factors, such as pose in the scene, shape of the object, illumination condition and reflectance properties, are coupled to affect the appearance of the object. Thus, it is difficult for the monocular vision to estimate the pose of the object. In this paper, we present a novel solution for object pose estimation with a monocular vision.

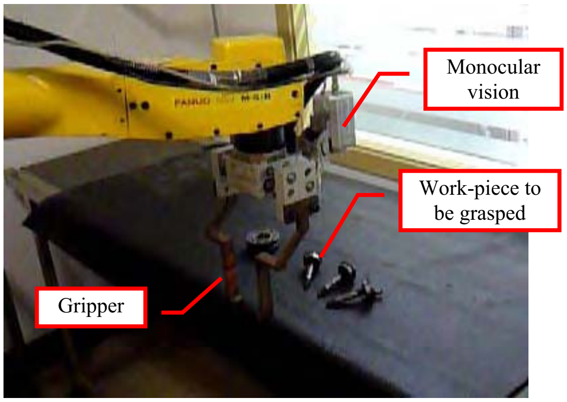


Fig. 1 Illustration of robotic object grasping with a monocular vision.

III. Bilinear Model Based Solution to Object Pose Estimation with Monocular Vision

The roll angle of the object pose can be ignored by considering that most industrial objects are symmetric. In the following, we will present a bilinear model based solution to obtain the rest five parameters of the object pose.

As shown in Fig. 2, the proposed method consists of three main steps:

S.1 Estimate the translation (x, y) of the object while detecting it by some computer vision algorithms, such as chamfer distance [18].

S.2 Segment the object from the original image and then estimate its yaw angle θ and pitch angle φ by factorizing the object appearance into the yaw factor and the pitch factor with the constructed bilinear model.

S.3 Estimate the translation z of the object center using the obtained pitch angle φ and the geometry information of the object.

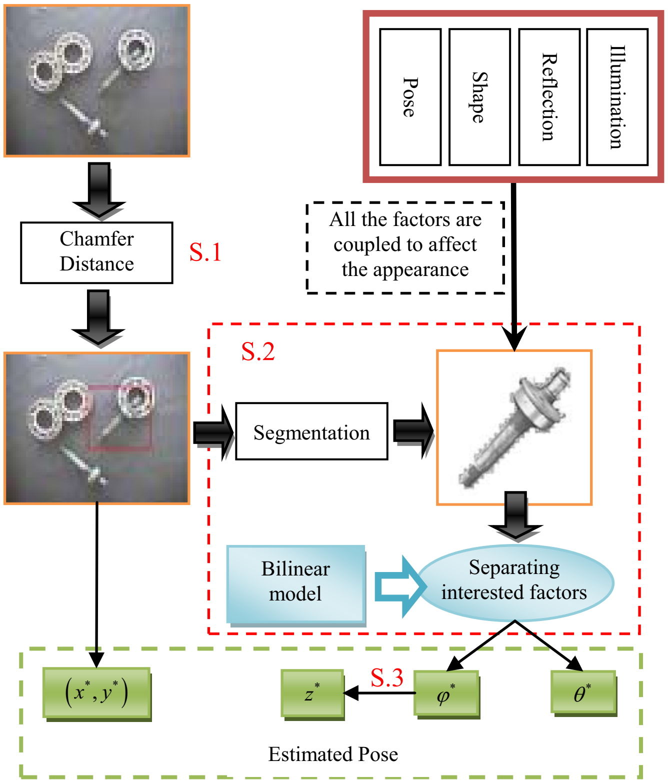


Fig. 2 Bilinear Model Based Solution to Object Pose Estimation. 1) A chamfer distance algorithm [18] is used to locate the object and estimate its position. 2) Then, the object is segmented from the image with some image templates; and the pitch angle and the yaw angle is estimated by a constructed bilinear model. 3) Finally, translation z of the object center is estimated by using the geometry information and the obtained pitch angle.

It is noted that the location method is not restricted, and any suitable algorithm can be used in the proposed solution. In our pose estimation system, the chamfer distance algorithm [18] is adopted for object detection. The detection algorithm is introduced briefly. Denote the templates of the object to be matched by $I_t, t = 1, 2, \cdots, T$ and the image to be detected is I_0. A series of sub-images $I_s, s = 1, 2, \cdots, S$ with the same size as I_t are extracted from I_0 successively. Then, the object can be detected by computing the minimum similarity of the Hausdorff Distances:

$$D_{HD} = \min_{s,t} H\left(I_t, I_s\right), \qquad (1)$$

where $H\left(I_t,I_s\right)=\max\left\{h\left(I_t,I_s\right),h\left(I_s,I_t\right)\right\}$ and the function $h\left(A,B\right)$ is defined as $h\left(A,B\right)=\max\limits_{a\in A}\min\limits_{b\in B}\left\|a-b\right\|$.

Since the subscript s of I_s implies the position of the detected object in the original image I_0, the translations x^* and y^* can be obtained when the object is detected.

Denote the image template and the sub-image which make (1) achieve its minimum by I_t^* and I_s^*, respectively. Using I_t^*, the sub-image I_s^* is segmented to I_g with its background wiped off (see Fig. 2).

Then, a bilinear model, which is constructed offline with training data, is applied to the segmented image I_g to separate the yaw factor and the pitch factor of the pose. Then, the yaw angle θ^* and the pitch angle φ^* are estimated by comparing the computed parameters with the model parameters. Details are described in the IV part.

With the pitch angle obtained from the bilinear model and the prior geometry information of the object, the translation z^* of the object center can be computed.

IV. Bilinear Model Construction and Object Pose Estimation

To estimate the object pose from a single view is difficult because the depth information is lost in the monocular vision system. What's worse, a number of factors, such as pose in the scene, shape of the object, illumination condition and reflectance properties, is coupled to affect the appearance of the object. Thus, the separation of the factors, such as the pitch angle and the yaw angle, is useful for estimating the pose of the object.

The bilinear model, proposed by Tenenbaum and Freeman [9], is good at separating the observations (object appearance) into two independent factors as so called style and content (Fig. 3). In this work, we are interested in separating the yaw angle and the pitch angle from the object appearance image.

A. Bilinear Model Construction

In this section, the construction of the bilinear model is described by considering the pitch angle as a style factor and the yaw angle as a content factor.

Given an image I^{py} of the object located at the pose with the pitch angle φ_p and the yaw angle θ_y, we can stack it into a vector $\mathbf{g}^{py}$. In order to reduce the dimension of the data, we apply a random projection [17] to the image vector $\mathbf{g}^{py}$:

$$\mathbf{z}^{py}=\mathbf{R}\cdot\mathbf{g}^{py}, \tag{2}$$

where $\mathbf{R}$ denotes a $K\times N$ random projection matrix and $\mathbf{z}^{py}$ denotes the K dimensional vector for the input image after dimensionality reduction. The dimension K is appointed according to the application while N is the original dimension. The entries $\left\{r_{kn}\right\}$ of the matrix $\mathbf{R}$ are defined as: $r_{kn}\sim N\left(0,1\right)$, $\left\|\mathbf{r}_k\right\|=1$.

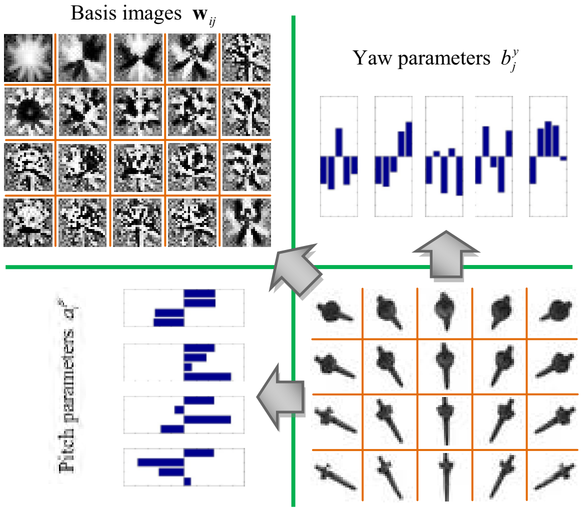


Fig. 3 Illustration of the bilinear model for a small set of axles. 20 appearance images of the axle with 4 different pitch angles and 5 different yaw angles are decomposed into pitch parameters a_i^p, yaw parameters b_j^y and basis images $\mathbf{w}_{ij}$. To render an object image in a particular pitch angle and a particular yaw angle, the basis images is linearly combined with coefficients given by corresponding pitch and yaw parameters.

Then, each element z_k^{py} of $\mathbf{z}^{py}$ is given by the bilinear model as:

$$z_k^{py}=\sum_{i=1}^{I}\sum_{j=1}^{J}w_{ijk}a_i^p b_j^y=\left(\mathbf{a}^p\right)^{\mathrm{T}}\cdot\mathbf{W}_k\cdot\mathbf{b}^y, \tag{3}$$

where $\mathbf{a}^p$ is the I dimensional pitch parameter vector, $\mathbf{b}^y$ is the J dimensional yaw parameter vector, a_i^p is the ith element of $\mathbf{a}^p$, b_j^y is the jth element of $\mathbf{b}^y$. $\mathbf{W}_k$ is a matrix models the interaction of the pitch and yaw factors, and w_{ijk} are the elements of $\mathbf{W}_k$. We rewrite the aforementioned equation in a matrix form:

$$\mathbf{z}^{py}=\sum_{i=1}^{I}\sum_{j=1}^{J}\mathbf{w}_{ij}a_i^p b_j^y, \tag{4}$$

where $\mathbf{w}_{ij}=[w_{ij1},w_{ij2},\cdots,w_{ijK}]^T$ are stacked basis vectors (images) of dimension K. (4) shows that the object observation vector $\mathbf{z}^{py}$ can be generated by mixing these basis vectors $\mathbf{w}_{ij}$ with the coefficients given by the pitch parameters a_i^p and the yaw parameters b_j^y. Fig. 3 illustrates how the bilinear model decomposes the object appearances into pitch parameters a_i^p and yaw parameters b_j^y, and the basis vectors $\mathbf{w}_{ij}$ describe their interactions.

Suppose T poses of the specific object is observed with T_p pitch angles and T_y yaw angles ($T=T_p\cdot T_y$). To fit the bilinear model is to find the yaw parameters a_i^p, the pitch parameters b_j^y and the interaction coefficients w_{ijk}. According to (2), the model parameters can be obtained by minimizing the total square error [9]:

$$E = \sum_{p=1}^{T_p} \sum_{y=1}^{T_y} \sum_{k=1}^{K} \left(z_k^{py} - \left(\mathbf{a}^p \right)^{\mathrm{T}} \cdot \mathbf{W}_k \cdot \mathbf{b}^y \right)^2 . \tag{5}$$

To fix this problem, we follow Tenenbaum and Freeman [9] to use the singular vector decomposition algorithm to estimate the model parameters. All the training data are firstly stacked into a $\left(T_p \times K \right) \times T_y$ matrix as follows:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}^{11} & \cdots & \mathbf{z}^{1T_y} \\ \vdots & \ddots & \vdots \\ \mathbf{z}^{T_p 1} & \cdots & \mathbf{z}^{T_p T_y} \end{bmatrix}, \quad \mathbf{Z}^{\mathrm{VT}} = \begin{bmatrix} \mathbf{z}^{11} & \cdots & \mathbf{z}^{T_p 1} \\ \vdots & \ddots & \vdots \\ \mathbf{z}^{1T_y} & \cdots & \mathbf{z}^{T_p T_y} \end{bmatrix}, \tag{6}$$

where the superscript VT stands for vector transpose [19], each element $\mathbf{z}^{py}$ is a K dimensional vector for a particular object pose after dimensionality reduction. Then, the symmetric bilinear model is rewritten in a compact form as

$$\mathbf{Z} = \left(\mathbf{W}^{\mathrm{VT}} \mathbf{A} \right)^{\mathrm{VT}} \mathbf{B}, \ \mathbf{Z}^{\mathrm{VT}} = \left(\mathbf{W} \mathbf{B} \right)^{\mathrm{VT}} \mathbf{A}, \tag{7}$$

where $\mathbf{A}$ is the stacked pitch angle parameter matrix whose size is $I \times T_p$, $\mathbf{B}$ is the stacked yaw angle parameter matrix whose size is $J \times T_y$, and $\mathbf{W}$ is the stacked interaction matrix:

$$\mathbf{A} = \left[\mathbf{a}^1, \cdots, \mathbf{a}^{T_p} \right], \mathbf{B} = \left[\mathbf{b}^1, \cdots, \mathbf{b}^{T_y} \right], \mathbf{W} = \begin{bmatrix} \mathbf{w}_{11} & \cdots & \mathbf{w}_{1J} \\ \vdots & \ddots & \vdots \\ \mathbf{w}_{I1} & \cdots & \mathbf{w}_{IJ} \end{bmatrix} \tag{8}$$

To find the optimal model parameters, the SVD algorithm is applied iteratively until a convergence is achieved. The overall procedure of the model fitting is given below:

Algorithm I. Bilinear Model Fitting Algorithm

```
(1) Decompose the matrix $\mathbf{Z}$ with SVD algorithm into $\mathbf{Z} = \mathbf{USV}^{\mathrm{T}}$, then the
    stacked yaw parameter matrix $\mathbf{B}$ is estimated by the first $J$ rows of $\mathbf{V}^{\mathrm{T}}$.
(2) Compute $\left(\mathbf{ZB}^{\mathrm{T}}\right)^{\mathrm{VT}} = \mathbf{W}^{\mathrm{VT}}\mathbf{A}$. Apply SVD algorithm to $\left(\mathbf{ZB}^{\mathrm{T}}\right)^{\mathrm{VT}}$, the
    stacked pitch angle parameter matrix $\mathbf{A}$ is estimated by the first $I$ rows of $\mathbf{V}^{\mathrm{T}}$.
(3) Compute $\left(\mathbf{Z}^{\mathrm{VT}}\mathbf{A}^{\mathrm{T}}\right)^{\mathrm{VT}} = \mathbf{WB}$. Apply SVD algorithm to $\left(\mathbf{Z}^{\mathrm{VT}}\mathbf{A}^{\mathrm{T}}\right)^{\mathrm{VT}}$, the
    stacked pitch angle parameter matrix $\mathbf{B}$ is estimated by the first $J$ rows of $\mathbf{V}^{\mathrm{T}}$.
(4) Repeat steps (2) and (3) until the parameter matrix $\mathbf{A}$ and $\mathbf{B}$ converge.
(5) Compute the interaction matrix $\mathbf{W}$ by $\mathbf{W} = \left(\left(\mathbf{YB}^{\mathrm{T}}\right)^{\mathrm{VT}}\mathbf{A}^{\mathrm{T}}\right)^{\mathrm{VT}}$.  □
```

Then, the bilinear model is constructed; and the pitch parameters a_i^p and the yaw parameters b_j^y are fitted.

B. Pitch and Yaw Estimation with the Bilinear Model

After constructing the bilinear model and fitting its parameters, the pitch and yaw parameters of the object in a given image $\mathbf{z}^*$ can be computed using the training model parameters w_{ijk}. Denote the pseudo inverse of the matrix X by X^+, algorithm for computing pitch parameter vector $\mathbf{a}^*$ and yaw parameter vector $\mathbf{b}^*$ of the given image $\mathbf{z}^*$ is given as follows:

Algorithm II. Pitch and Yaw Parameters Estimation Algorithm

```
(1) Initial yaw parameter vector $\mathbf{b}^*$ with the mean vector of $\mathbf{B}$.
(2) Update the pitch parameter vector $\mathbf{a}^*$ by computing
    $\mathbf{a}^* = \left(\left(\mathbf{Wb}^*\right)^{\mathrm{VT}}\right)^+ \cdot \mathbf{z}^*$.
(3) Update the yaw parameter vector $\mathbf{b}^*$ by computing
    $\mathbf{b}^* = \left(\left(\mathbf{W}^{\mathrm{VT}}\mathbf{a}^*\right)^{\mathrm{VT}}\right)^+ \cdot \mathbf{z}^*$.
(4) Repeat steps (2) and (3) until the parameter vector $\mathbf{a}^*$ and $\mathbf{b}^*$ converge.
    □
```

The particular pitch angle φ^* is identified by computing the minimum Euclidian distance between the computed pitch parameter vector $\mathbf{a}^*$ and each training pitch parameter vector $\mathbf{a}^p$ as follows:

$$p_{opt} = \arg_p \min \left\| \mathbf{a}^p - \mathbf{a}^* \right\|, \tag{9}$$

Similarly, the particular yaw angle θ^* is identified as follows:

$$y_{opt} = \arg_y \min \left\| \mathbf{b}^y - \mathbf{b}^* \right\|. \tag{10}$$

However, identifying the factors by comparing Euclidian distances between the computed parameters and the fitted parameters does not consider the fact that different elements account for different importance. Thus, we present a scheme to enhance the robustness of the factor identification by adjusting the weights of the parameters according to their importance to the appearance.

It is noticed that the elements a_i^p and b_j^y of the model parameter vectors are arranged in a descending order according to their contributions to the pitch angle and the yaw angle, respectively. Furthermore, it is found out that the first two elements of the parameter vectors always play a critical role in factor identification. Thus, different weights are set to the elements as:

$$u_i = \begin{cases} \beta L, & i = 1,2 \\ L - i + 1, & i = 3, \cdots, L \end{cases} , \tag{11}$$

where $\beta > 1$ and L is the length of the parameter vector. Then, the pitch angle is identified by computing the minimum weighted Euclidian distance between the computed pitch parameters and the fitted pitch parameters as follows:

$$p_{opt} = \arg_p \min \sqrt{\sum_{i=1}^{I} u_i^2 \left(a_i^p - a_i^* \right)^2}, \ p \in \left\{ 1, 2, \cdots, M_p \right\} \tag{12}$$

Similarly, the yaw angle is identified as:

$$y_{opt} = \arg_y \min \sqrt{\sum_{j=1}^{J} u_j^2 \left(b_j^y - b_j^* \right)^2}, \ y \in \left\{ 1, 2, \cdots, M_y \right\} \tag{13}$$

V. Experiment Results and Discussion

In this section, an axle is taken as an example to illustrate the proposed solution and validate its efficiency. In our work, a database consists of 840 images of the axle has been built up. The images were taken by the 704×576 CCD camera mounted on the industrial robot. The axle was placed at 35 different yaw angles θ as: $\theta = -85°, -80°, \cdots, 80°, 85°$. For each yaw angle, the axle was fixed at 24 different pitch angles as follows: $\varphi = 0°, 5°, 7.5°, 10°, \cdots, 57.5°, 60°$.

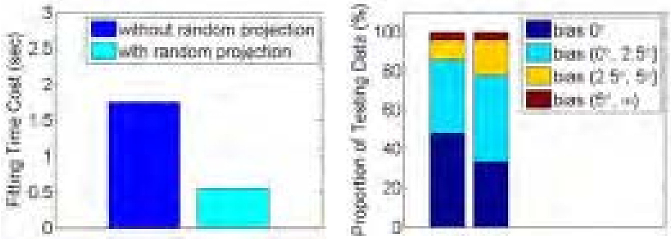
We clipped a 240×240 sub-image, which contained the axle located at the center position, from each original image. Then, the axle was segmented from the background manually.

A. Performance of Bilinear Model with Random Projection

The key skill used in the bilinear model fitting is singular vector decomposition (SVD) which suffers computation cost when the dimension of the data is large (e.g. be of dimension 5000). Thus, the dimension of data is reduced by using a random projection algorithm. To evaluate the validity of this dimensionality reduction, an experiment is done.

First, 195 images are taken from the data base with 15 pitch angles as $\varphi = 10°, 12.5°, \cdots, 45°$ and 13 yaw angles as $\theta = -30°, -25°, \cdots, 25°, 30°$. All the images are cropped to the size of 24×24. 105 images ($\varphi = 10°, 12.5°, \cdots, 45°$ and $\theta = -30°, -20°, \cdots, 20°, 30°$) are taken for training while the rest 90 images for testing.

Two copies of the images are used in the experiment. One is for the bilinear model without random projection; the other is for the bilinear model with the dimension reducing to 128. Fig. 4 shows the performance comparison between the two methods.



(a) (b)

Fig. 4 Performances of the bilinear model with and without random projection.(a) shows the time cost of the model fitting with random projection (the right one) is only one third of that without random projection (the left one). (b) shows that compared with the original model (the left column), the model with random projection (the right column) has a near performance in pitch angle estimation.

In this comparative experiment, the fitting time of the model with random projection is only one third of that without dimensionality reduction. The yaw angles are identified with the same accuracy after dimensionality reduction; and the model with random projection has nearly the same performance as the original one in pitch identification (Fig. 4 (b)).

Therefore, the random projection keeps the performance of the bilinear model while reducing the dimension of the data and the computation time.

B. Estimation Performance of Pitch Angle and Yaw angle

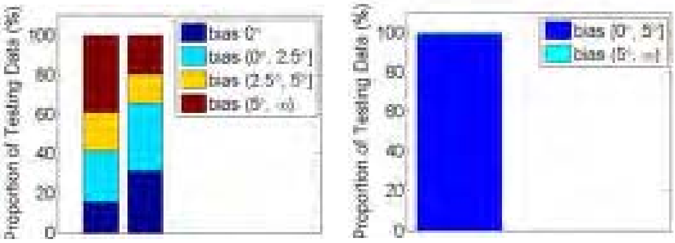
Among the 840 images, 432 images (with 18 yaw angles who are 10 degrees apart and 24 pitch angles for each yaw angle) are chosen for model fitting, while the others are used for testing.

The images for model fitting are with yaw angles as $\theta = -85°, -75°, \cdots, 75°, 85°$ and with the pitch angles as $\varphi = 0°, 5°, 7.5°, 10°, \cdots, 57.5°, 60°$.

The images for pose estimation testing are with the yaw angles as $\theta = -80°, -70°, \cdots, 70°, 80°$ and with the pitch angles as $\varphi = 0°, 5°, 7.5°, 10°, \cdots, 57.5°, 60°$.

Training the chosen data through algorithm I, the yaw parameter vectors $\mathbf{b}^y$, $y = 1, 2, \cdots, 18$, the pitch parameter vectors $\mathbf{a}^p$, $p = 1, 2, \cdots, 24$ and the interaction matrix $\mathbf{W}$ are obtained.

For each testing image in the data base, we compute its pitch parameter vector $\mathbf{a}^*$ and yaw parameter vector $\mathbf{b}^*$ using algorithm II. Then, the pitch angle and the yaw angle are identified with the propose scheme. Fig. 5 shows the performance of the angle identification by using bilinear model.



(a) Pitch angle estimation (b) Yaw angle estimation

Fig. 5 Performances of pitch and yaw angle estimation by using bilinear model. (a) shows the estimation performance of the pitch angle. The left column is the performance of angle estimation for angles which are less than 20 degree; and the right one is for angles which are larger than 20 degrees. (b) shows the estimation performance of the yaw angle. It can be seen that nearly all the angles estimated are within the 5 degrees.

The pitch angles, which are used for model fitting, are with 2.5 degrees apart. Experiments show that the pitch estimation is much more accurate when it is larger than 20 degree. The reason is that the model parameter estimation is sometimes trapped into a local minimum instead of the global minimum when the pitch angle is less than 20 degree.

The yaw angles for model fitting are 10 degrees apart from each other. It is noticed that the yaw angles of the testing images are not in the training set but in the interpolation positions of the training yaw angles. Thus, the accuracy of the yaw angle estimation is 5 degrees. Experiment show that nearly all the testing images get the accurate yaw angle.

C. Object Grasping by Using the Proposed Solution

The proposed pose estimation solution is tested on the robotic axle grasping task. A monocular vision guiding axle-bearing assembly system is built in our lab.

Considering the characteristics of the axle, a Hough transform [20] is used to find the coarse yaw angle $\tilde{y}$ first. Then 5 templates with different pitch angles at the $\tilde{y}$ angle are taken from the axle image data base. The chamfer distance is used to detect the axle in the captured image with the chosen templates. After the axle is detected, a sub-image I_s^* containing the axle is clipped from the original image.

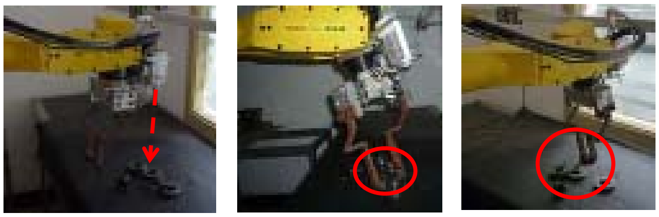
Although the fitted template implies the coarse pitch angle $\tilde{p}$ and yaw angle $\tilde{y}$, they have to be re-estimated further using the bilinear model. 15 templates I_t^{py} are taken from the data base for segmentation. The templates are as follows: I_t^{py}, where $p = \tilde{p}-2, \tilde{p}-1, \tilde{p}, \tilde{p}+1, \tilde{p}+2$, $\tilde{y} = \tilde{y}-1, \tilde{y}, \tilde{y}+1$. The segmentation is as:

$$I_{seg}^{py}(u,v) = \begin{cases} I_s(u,v), & I_t^{py}(u,v) \neq 255 \\ 255, & I_t^{py}(u,v) = 255 \end{cases}, \tag{14}$$

For each segmented image I_{seg}^{py}, we separate it into the pitch factor and the yaw factor, and find the corresponding pitch and yaw parameters by the propose method. The corresponding pitch and yaw parameters with the minimum weighted Euclidian distance are chosen to estimate the pitch angle and the yaw angle.

Finally, the height of the object center from the conveyer belt can be computed by using the estimated pitch angle and the geometry information of the axle.

Fig. 6 shows the experiment that the robot grasps the axle using the proposed pose estimation solution.



(a) (b) (c)

Fig. 6 Performance of pitch and yaw angle estimation by using bilinear model. (a) the image of the axle is captured by the CCD camera; and the pose of the axle is estimated by the proposed solution. (b) and (c) shows the axle grasping process.

D. Discussion

It should be pointed out that although bilinear model is used for separating the pitch and the yaw angle in this work, it is also suitable for separating any two independent factor of the object observation, e.g. the translations on the plane.

One of the advantages of this factor separation method is that the geometry model of the object is unknown. Thus, it can be used for pose estimation of other object.

The angle intervals of the pitch angle and the yaw angle in the experiment are both 5 degree, but more experiments will be done in a more detailed on in our future work and the maximal bias of the method will be examined.

VI. Conclusion

In the paper, a bilinear model, which separates the pitch factor and the yaw factor from the object appearance, is used to estimate the pitch and the yaw angles. In order to tackle the computation in model fitting, a random projection is applied to the data while preserving its efficiency. To improve the robustness of the factor identification, a weighted Euclidian distance based method is proposed. Based on the bilinear model for estimating the pitch and yaw angle, a three-step object pose estimation solution is given. Experiments show the validity of the proposed pose estimation solution.

References

[1] J. Batlle, E. Mouaddib and J. Salvi, “Recent progress in coded structured light as a technique to solve the correspondence problem: a survey,” *Pattern Recognition*, vol. 31, no. 7, pp. 963-982, Jul. 1998.

[2] D. Scharstein and R. Szeliski, “High-accuracy stereo depth maps using structured light,” In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.1, pp. 195-202, Jun. 2003.

[3] D. Breitenreicher and C. Schnörr, “Model-based multiple rigid object detection and registration in unstructured range data,” *International Journal of Computer Vision*, vol. 92, no. 1, pp. 32-52, Mar. 2011.

[4] M. Y. Liu, et al, “Pose estimation in heavy clutter using a multi-flash camera,” In *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2028-2035, May 2010.

[5] R. M. Haralick, et al, “Pose estimation from corresponding point data,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 6, pp. 1426-1446, 1989.

[6] D. F. Dementhon and L. S. Davis, “Model-based object pose in 25 lines of code,” *International Journal of Computer Vision*, vol. 15, no. 1-2, pp. 123-141, 1995.

[7] A. Ansar and K. Daniilidis, “Linear pose estimation from points or lines,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 578-589, May 2003.

[8] H. Murase and S. K. Nayar, “Visual learning and recognition of 3-D objects from appearance,” *International Journal of Computer Vision*, vol. 14, no. 1, pp. 5-24, 1995.

[9] J. B. Tenenbaum and W. T. Freeman, “Separating style and content with bilinear models,” *Neural Computation*, vol. 12, no. 6, pp. 1247-1283, Jun. 2000.

[10]D. Shin, H. S. Lee and D. Kim, “Illumination-robust face recognition using ridge regressive bilinear models,” *Pattern Recognition Letters*, vol. 29, no. 1, pp. 49-58, Jan. 2008.

[11]X. B. Gao and C. N. Tian, “Multi-view face recognition based on tensor subspace analysis and view manifold modeling,” *Neurocomputing*, vol. 72, no. 16-18, pp. 3742-3750, Oct. 2009.

[12]I. Mpiperis, S. Malassiotis and M. G. Strintzis, “Bilinear models for 3-D face and facial expression recognition,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 498-511, Sep. 2008.

[13]C. Zhou and X. Y. Lin, “Facial expressional image synthesis controlled by emotional parameters,” *Pattern Recognition Letters*, vol. 26, no. 16, pp. 2611-2627, Dec. 2005.

[14]D. B. Grimes and R. P. N. Rao, “Bilinear sparse coding for invariant vision,” *Neural Computation*, vol. 17, no. 1, pp. 47-73, Jan. 2005.

[15]A. Elgammal and C. S. Lee, “Separating style and content on a nonlinear manifold,” In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 478-485, 2004.

[16]H. J. Song and H. S. Kim, “Bilinear model-based maximum likelihood linear regression speaker adaptation framework,” *Signal Processing Letters*, vol. 16, no. 12, pp. 1063 - 1066, Dec. 2009.

[17]S, Kaski, “Dimensionality reduction by random mapping: fast similarity computation for clustering,” in *IEEE International Joint Conference on Neural Networks*, vol. 1, pp. 413–418, 1998.

[18]G. Borgefors, “Hierarchical chamfer matching: a parametric edge matching algorithm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 849-865, Nov. 1988.

[19]D. H. Marimont and B. A. Wandell, “Linear models of surface and illuminant spectra,” *Journal of the Optical Society of America A*, vol. 9, no. 11, pp. 1905-1913, Nov. 1992.

[20]D.H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, 1981.