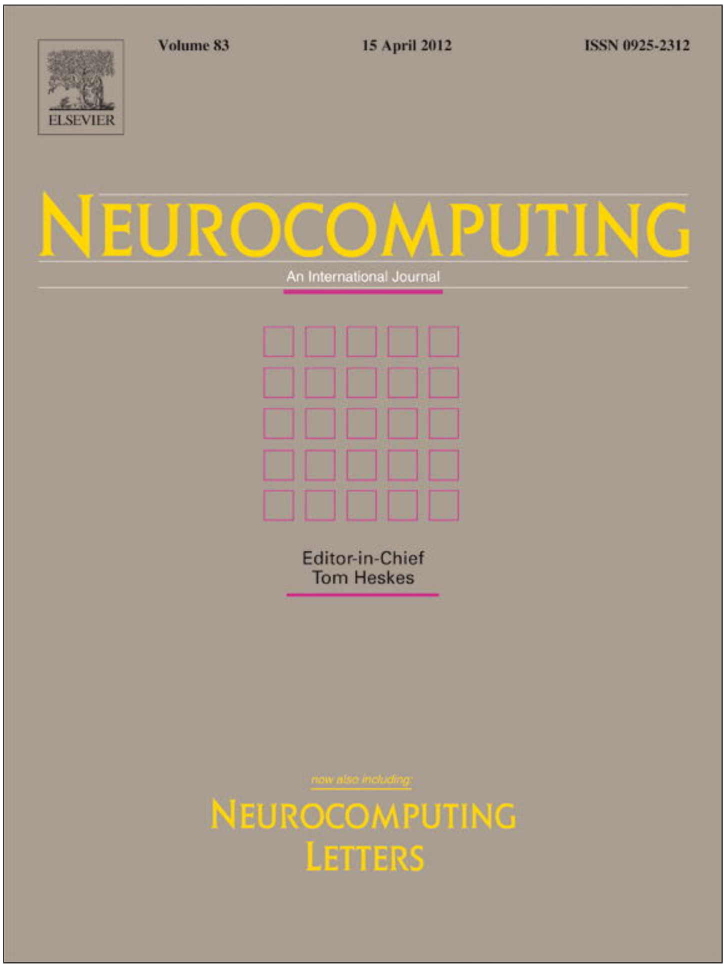


Volume 83 15 April 2012 ISSN 0925-2312

ELSEVIER

NEUROCOMPUTING
An International Journal

Editor-in-Chief
Tom Heskes

now also including:
NEUROCOMPUTING
LETTERS

Neurocomputing 83 (2012) 176–187

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Sub-pattern bilinear model and its application in pose estimation of work-pieces

Zhicai Ou*, Peng Wang, Jianhua Su, Hong Qiao

*Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

ARTICLE INFO

*Article history:*
Received 15 August 2011
Received in revised form
28 November 2011
Accepted 11 December 2011
Communicated by X. Gao
Available online 28 December 2011

*Keywords:*
Bilinear models
Sub-pattern bilinear model
Pose estimation
Work-piece
Occlusion

ABSTRACT

Bilinear models have been proposed to separate the factors from the observations for joint factors identification or translation tasks.

However, the performance of existing bilinear models may degrade under challenging conditions when local image information cannot be obtained caused by occlusions or image noises. In this paper, a novel sub-pattern bilinear model (SpBM) is proposed. Different from existing bilinear models, SpBM constructs the sub-pattern bilinear model through a novel learning algorithm utilizing local patterns generated by dividing global patterns in a deterministic way. As a result, the specific factors of testing observation are identified by synthesizing the discriminative information provided by the local sub-patterns.

To further improve the identification performance of SpBM, a new ridge regressive parameter estimation algorithm (RRPE) is also proposed. RRPE introduces the ridge regression into parameter estimation to stabilize the matrix inverse computation and alleviate the non-convergent cases.

The proposed sub-pattern bilinear model is introduced into pose estimation of work-pieces to separate and estimate some key pose factors individually. Experimental results demonstrate the effectiveness of the proposed method.

## 1. Introduction

Observations in perceptual systems, such as appearances of objects or voices of speakers, are commonly an interaction of different independent physical factors, for instance, handwriting styles and characters [1], face identities and viewpoints [2], localized features and transformations [3], person identities, expressions, head poses and illuminations [4], and so on. How to separate the factors from the observations is still an open question for joint factors identification or translation.

According to the routine separation requirements of perceptual systems, Tenenbaum and Freeman [1] proposed bilinear models which separate the independent style and content factors of the observations. Bilinear models are linear in either factor when the other is held constant. Thus, they share almost all of the advantages of linear models: they have simple structures and are easy to implement; they can be trained by efficient well-known techniques, such as the singular value decomposition (SVD) and the expectation-maximization algorithm (EM) [1,13].

Some extensions and improvements of bilinear models have been proposed by other researchers. Vasilescu and Terzopoulos [4] gave a multi-linear model based on tensor theory. In their work, the observations of the faces are decomposed into orthogonal factors including person identities, expressions, head poses and illuminations; and the interaction of these multi-factors are modeled by the basic images, i.e., tensorfaces. Elgammal and Lee [5] embedded bilinear models in a nonlinear manifold and separated the intrinsic body configuration from the appearance of the person for gait recognition. Du and Lin [6] proposed a general bilinear model in kernel space by mapping the observations into a kernel space. Similarly, Zhou and Lin [7] used kernel-based bilinear models to decompose the identity and expression of face image, and then synthesized the realistic comprehensive expressional images by controlling the decomposed identity and expression parameters. Gonzalez-Mora et al. [8] proposed bilinear active appearance models to separate pose from expression/identity changes within the active appearance model framework.

Although different improvements have been made to bilinear models, there are still some issues that should be further dealt with. Firstly, existing bilinear models usually separate style and content from observations using global patterns, such as the whole appearance of faces, and ignore the description of local image information. Therefore, the performance of these methods may degrade under challenging conditions when local image information cannot be obtained caused by occlusions or image noises. Secondly, parameter estimation in bilinear models for

* Corresponding author. Tel.: +86 10 82610027.
*E-mail address:* zhicai.ou@ia.ac.cn (Z. Ou).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.12.012

separating the style and the content from testing observations always require a repetitive computation of matrix inverse operations [1]. This may lead to a non-convergent case which consequently affects the identification or translation results.

To overcome these limitations, in this paper, a novel sub-pattern bilinear model (SpBM) combined with a new ridge regressive parameter estimation algorithm (RRPE) is proposed. The sub-pattern bilinear model is constructed utilizing local patterns, and then the style and content factors are separated from observations with a new ridge regressive parameter estimation algorithm. This extends the ability of existing bilinear models in challenging cases when local image information cannot be obtained, because the rest sub-patterns are usually able to provide enough discriminative information for identification.

In the proposed method, firstly, the original observations are divided into some smaller non-overlapped sub-observations in a deterministic way. All the sub-observations sharing the same sub-pattern are collected to construct the sub-pattern training sets. Secondly, a sub-bilinear model training algorithm is designed to estimate the model parameters using the sub-pattern training sets. Thirdly, the style and content parameters of the testing observation are estimated using the proposed ridge regressive parameter estimation algorithm with a novel objective function. Finally, the obtained style and content parameters of the sub-patterns are synthesized with majority voting and nearest neighbor algorithm to give the final identification results.

The proposed sub-pattern bilinear model is introduced into pose estimation of work-pieces to separate and estimate some key pose factors individually. SpBM shows satisfactory results under challenging conditions, such as occlusions and noises.

The main contributions of this paper are summarized in three aspects:

(1) A novel sub-pattern bilinear model (SpBM) is proposed. Different from existing bilinear models, SpBM constructs the sub-pattern bilinear model through a novel learning algorithm utilizing local patterns generated by dividing global patterns in a deterministic way. As a result, the specific style and content of testing observations are identified by synthesizing the discriminative information provided by the local sub-patterns. Thus, SpBM is able to obtain good identification results in the cases when local image information cannot be obtained caused by occlusions or image noises.

(2) To further improve the identification performance of SpBM, a new ridge regressive parameter estimation algorithm (RRPE) is also proposed. RRPE introduces the ridge regression into parameter estimation to stabilize the matrix inverse computation and alleviate the non-convergent cases. And some theoretical analysis of RRPE is also given to prove its efficiency.

(3) The proposed sub-pattern bilinear model is introduced into pose estimation of work-pieces to separate and estimate some key pose factors individually. Experimental results demonstrate the effectiveness of the proposed method.

The remainder of this work is organized as follows: we review existing bilinear models briefly in Section 2. In Section 3, the development of sub-pattern bilinear model is described. The generation of sub-patterns, the training algorithm of SpBM, the new ridge regressive parameter estimation and the identification algorithm are presented. Experiments which utilize SpBM to separate and estimate some key pose factors of work-pieces from simulated and real images are given in Section 4. Section 5 gives the conclusions.

## 2. Review of bilinear models

Bilinear models which separate and identify two independent factors (such as style and content) are firstly proposed by Tenenbaum and Freeman [1]. Fig. 1 shows the overview of existing bilinear models. In the learning stage, the style factor and the content factor, represented by parametric vectors, are estimated by alternately updating one factor within each iteration until convergence; and then the interaction matrix are derived from the estimated style and content factors (Algorithm 1). In the identification stage, the style and content factors of the testing observation are separated using the interaction matrix obtained in the learning stage (Algorithm 2); and then the specific style and content are identified by comparing with the model parameters.

### 2.1. Bilinear model learning

Let $\mathbf{z}^{sc}$ denote a $K$ dimension observation vector in specific style $s$ and content $c$. It is assumed that $\mathbf{z}^{sc}$ is given by the general bilinear form [1]:

$$\mathbf{z}^{sc} = \sum_{i=1}^{I}\sum_{j=1}^{J} \mathbf{w}_{ij}\mathbf{a}_i^s\mathbf{b}_j^c \tag{1}$$

where the observation $\mathbf{z}^{sc}$ is separated into two independent factor vectors $\mathbf{a}^s = (a_1^s, a_2^s, \ldots, a_I^s)^{\mathrm{T}}$ and $\mathbf{b}^c = (b_1^c, b_2^c, \ldots, b_J^c)^{\mathrm{T}}$, with a series of interaction vectors $\mathbf{w}_{ij}$ of dimension $K$. $I$ and $J$ are the dimensions of $\mathbf{a}^s$ and $\mathbf{b}^c$, respectively. Given a set of training observations $\{\mathbf{z}^{sc}\}_{s=1,c=1}^{T_s,T_c}$ of $T_s$ styles and $T_c$ contents, the model parameters $\{\mathbf{a}^s\}_{s=1}^{T_s}$, $\{\mathbf{b}^c\}_{c=1}^{T_c}$ and $\{\mathbf{w}_{ij}\}_{i=1,j=1}^{I,J}$ can be computed by minimizing the total squared error [1]:

$$E^{learn} = \sum_{s=1}^{T_s}\sum_{c=1}^{T_c} \left\| \mathbf{z}^{sc} - \sum_{i=1}^{I}\sum_{j=1}^{J} \mathbf{w}_{ij}\mathbf{a}_i^s\mathbf{b}_j^c \right\|^2, \tag{2}$$

We first stack all the training data into a matrix as follows [1]:

**Algorithm 1.** Bilinear model learning algorithm

(1) Decompose the matrix $\mathbf{Z}$ with SVD algorithm into $\mathbf{Z}=\mathbf{USV}^{\mathrm{T}}$, then the stacked content parameter matrix $\mathbf{B}$ is initialized by the first $J$ rows of $\mathbf{V}^{\mathrm{T}}$.

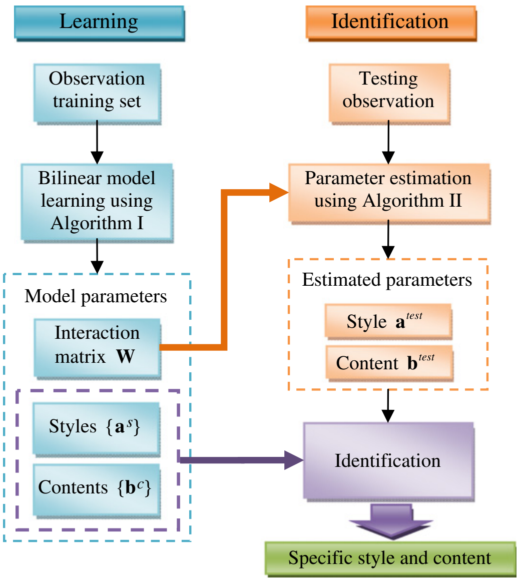

Learning
Identification
Observation training set
Testing observation
Bilinear model learning using Algorithm I
Parameter estimation using Algorithm II
Model parameters
Estimated parameters
Interaction matrix W
Style $\mathbf{a}^{test}$
Content $\mathbf{b}^{test}$
Styles $\{\mathbf{a}^s\}$
Contents $\{\mathbf{b}^c\}$
Identification
Specific style and content

**Fig. 1.** Overview of bilinear models.

**Table 1**
Comparison between Independent Component Analysis and Bilinear Models.

| | Independent Component Analysis | Bilinear Models |
|---|---|---|
| Purpose | Find a linear representation of statistically independent non-Gaussian components | Separate one factor (content) from the other (style) from the observations |
| Model | $\mathbf{x}=\sum_{i=1}^{n}\mathbf{a}_i s_i$ | $\mathbf{z}^{sc}=\sum_{i=1}^{I}\sum_{j=1}^{J}\mathbf{w}_{ij}\mathbf{a}_i^s\mathbf{b}_j^c$ |
| Model Category | Linear | Linear |
| Characteristics of the components or factors | • Components are of the same kind<br>• Statistically independent<br>• Non-Gaussian | • Two kinds of components according to the two factors<br>• Independent factors |
| Preprocessing | Centering and Whitening | None |
| Training method | fastICA | Singular value decomposition or Quasi-Newton method |
| Variances of the components or factors | Cannot be determined | Cannot be determined |
| Component order | Cannot be determined | In an descending order according to the singular values |
| Applications (not limited) | • Separation of artifacts in MEG data [25,26]<br>• Finding hidden factors in financial data [25,26]<br>• Reducing noise in natural images [25,26]<br>• Telecommunications [25,26]<br>• Face recognition [27]<br>• Voice signal processing [28]<br>• Image separation [29] | • Facial expression synthesis [1,7]<br>• Letter extrapolation [1]<br>• Sparse coding [3]<br>• Face/Facial recognition [10,11,13]<br>• Gait recognition [5,12]<br>• Speaker adaption [14]<br>• Spatio-temporal point distribution analysis [15] |

(2) Compute $(\mathbf{Z}\mathbf{B}^{\mathrm{T}})^{\mathrm{VT}}=\mathbf{W}^{\mathrm{VT}}\mathbf{A}$. Apply SVD algorithm to $(\mathbf{Z}\mathbf{B}^{\mathrm{T}})^{\mathrm{VT}}$, the stacked style parameter matrix $\mathbf{A}$ is estimated by the first $I$ rows of $\mathbf{V}^{\mathrm{T}}$.

(3) Compute $(\mathbf{Z}^{\mathrm{VT}}\mathbf{A}^{\mathrm{T}})^{\mathrm{VT}}=\mathbf{W}\mathbf{B}$. Apply SVD algorithm to $(\mathbf{Z}^{\mathrm{VT}}\mathbf{A}^{\mathrm{T}})^{\mathrm{VT}}$, the stacked content 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 matrices $\mathbf{A}$ and $\mathbf{B}$ converge.

(5) Estimate the interaction matrix $\mathbf{W}$ by computing:

$$\mathbf{W}=((\mathbf{Y}\mathbf{B}^{\mathrm{T}})^{\mathrm{VT}}\mathbf{A}^{\mathrm{T}})^{\mathrm{VT}}$$

$$\mathbf{Z}=\begin{bmatrix}\mathbf{z}^{11} & \cdots & \mathbf{z}^{1T_c}\\ \vdots & \ddots & \vdots\\ \mathbf{z}^{T_s1} & \cdots & \mathbf{z}^{T_sT_c}\end{bmatrix},\quad \mathbf{Z}^{\mathrm{VT}}=\begin{bmatrix}\mathbf{z}^{11} & \cdots & \mathbf{z}^{T_s1}\\ \vdots & \ddots & \vdots\\ \mathbf{z}^{1T_c} & \cdots & \mathbf{z}^{T_sT_c}\end{bmatrix} \tag{3}$$

where the superscript VT denote the vector transpose [1,9]. Then, the bilinear model can be rewritten in a compact form:

$$\mathbf{Z}=(\mathbf{W}^{\mathrm{VT}}\mathbf{A})^{\mathrm{VT}}\mathbf{B},\quad \mathbf{Z}^{\mathrm{VT}}=(\mathbf{W}\mathbf{B})^{\mathrm{VT}}\mathbf{A} \tag{4}$$

where $\mathbf{A}$ is the stacked style parameter matrix with size $I\times T_s$, $\mathbf{B}$ is the stacked content parameter matrix with size $J\times T_c$, and $\mathbf{W}$ is the stacked interaction matrix with size $(I\times K)\times J$. They can be represented as

$$\mathbf{A}=[\mathbf{a}^1,\ldots,\mathbf{a}^{T_s}],\quad \mathbf{B}=[\mathbf{b}^1,\ldots,\mathbf{b}^{T_c}],\quad \mathbf{W}=\begin{bmatrix}\mathbf{w}_{11} & \cdots & \mathbf{w}_{1J}\\ \vdots & \ddots & \vdots\\ \mathbf{w}_{I1} & \cdots & \mathbf{w}_{IJ}\end{bmatrix} \tag{5}$$

Based on the (4), the model parameters can be determined by an iterative singular value decomposition (SVD) algorithm [1] (Algorithm 1).

Convergence of Algorithm 1 can be guaranteed [1] (Magnus and Neudecker [23] proposed a proof for the scalar case, e.g. $K=1$; and this proof can be extended to the vector case considered here). Usually, Algorithm 1 converged within 5 steps in our experiments.

### 2.2. *Bilinear model parameter estimation*

Given a testing observation $\mathbf{z}^{test}$, its style and content factors are separated using the interaction matrix obtained in the training stage.

According to (1), the testing observation can be represented in a similar way as (4)

$$\mathbf{z}^{test}=(\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}}\mathbf{b}\ \text{or}\ \mathbf{z}^{test}=(\mathbf{W}\mathbf{b})^{\mathrm{VT}}\mathbf{a} \tag{6}$$

Then, the style and content parameters can be expressed as follows:

**Algorithm 2.** Parameter estimation algorithm

(1) Initialize the content parameter $\mathbf{b}$ with the mean vector of $\mathbf{B}$.

(2) Update the style parameter $\mathbf{a}$ by computing:

$$\mathbf{a}=((\mathbf{W}\mathbf{b})^{\mathrm{VT}})^{+}\cdot\mathbf{z}^{test}.$$

(3) Update the content parameter $\mathbf{b}$ by computing:

$$\mathbf{b}=((\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}})^{+}\cdot\mathbf{z}^{test}.$$

(4) Repeat steps (2) and (3) until the parameters $\mathbf{a}$ and $\mathbf{b}$ converge.

$$\mathbf{a}=((\mathbf{W}\mathbf{b})^{\mathrm{VT}})^{+}\cdot\mathbf{z}^{test} \tag{7}$$

$$\mathbf{b}=((\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}})^{+}\cdot\mathbf{z}^{test} \tag{8}$$

where $(\bullet)^{+}$ denotes the pseudo-inverse. (7) and (8) suggest an iterative algorithm to estimate the style and content parameters. By initializing the content parameter vector $\mathbf{b}$ as the mean vector of $\mathbf{B}$, the style and content parameters ($\mathbf{a}$ and $\mathbf{b}$) can be derived by alternately updating one factor within each iteration until convergence [1] (Algorithm 2). Algorithm 2 converges if the condition numbers satisfy: $\kappa((\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}})<\eta$ and $\kappa((\mathbf{W}\mathbf{b})^{\mathrm{VT}})<\eta$, where $\kappa(\mathbf{X})=\sigma_{max}(\mathbf{X})/\sigma_{min}(\mathbf{X})$, $\sigma_{max}(\mathbf{X})$ and $\sigma_{min}(\mathbf{X})$ are the maximum and minimum singular values of $\mathbf{X}$, respectively, $\eta$ is the threshold for avoiding being ill-conditioned.

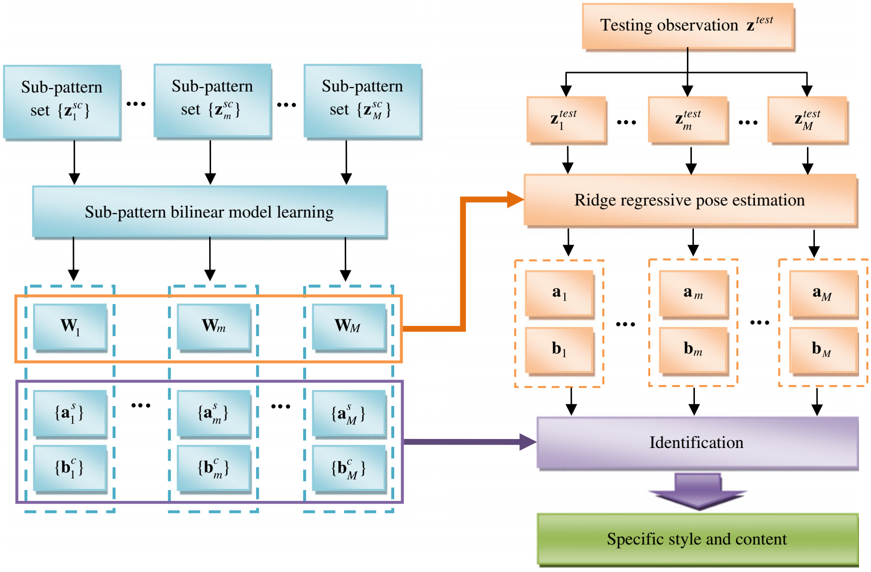


**Fig. 2.** Overview of the proposed sub-pattern bilinear model.

### 2.3. *Style and content identification*

The specific style and content of the testing observation can be identified in an exhausting searching manner. Denote the corresponding distances between the estimated and learned model parameters by

$$d_1(s,\mathbf{X}) = \|\mathbf{a}^s - \mathbf{X}\| \quad (9)$$

$$d_2(c,\mathbf{X}) = \|\mathbf{b}^c - \mathbf{X}\| \quad (10)$$

where $\mathbf{X}$ is the estimated parameter, $\mathbf{a}^s$ denotes the parameter of style $s$, $\mathbf{b}^c$ denotes the parameter of content $c$. The specific style and content are determined using the nearest neighbor classifier:

$$s^* = \arg_s \min d_1(s,\mathbf{a}) \text{ and } d_1(s^*,\mathbf{a}) < \theta_s \quad (11)$$

$$c^* = \arg_c \min d_2(c,\mathbf{b}) \text{ and } d_2(c^*,\mathbf{b}) < \theta_c \quad (12)$$

where $s^*$ and $c^*$ are the determined style and content, $\theta_s$ and $\theta_c$ are thresholds.

### 2.4. *Compared with ICA*

Independent Component Analysis (ICA) [25,26] is a linear transformation which represents the observation data by using non-Gaussian independent components. ICA shows a similar model with bilinear models as

$$\mathbf{x} = \sum_{i=1}^{n} \mathbf{a}_i s_i \quad (13)$$

Comparing (13) and (1), both ICA and bilinear models can be interpreted as an observation that is represented by a linear combination of the basic images ($\mathbf{w}_{ij}$ in bilinear models and $\mathbf{a}_i$ in ICA) with the weights provided by the components or factors.

However, there are some differences between the two methods. ICA obtains the independence of the components in ICA by decorrelating the high order statistics so that it seems to capture the essential structure of the data in many applications. In bilinear model, the “style” factor and the “content” factor are separated in the criterion of least square (e.g., second order statistics) so that observations in unfamiliar styles or contents can be identified or synthesized. And more differences are summarized in Table 1.

## 3. Sub-pattern bilinear model

Existing bilinear models usually separate style and content on global pattern of observations, and the performance of bilinear models mainly depends on the integrity of the observation data. Thus, bilinear models are lack of the capability to cope with challenging conditions when local image information cannot be obtained caused by occlusions or image noises. What's worse, there are usually repetitive computations of matrix inverse in Algorithm II which may lead to a non-convergent case. To overcome these limitations, a novel sub-pattern bilinear model (SpBM) combined with a ridge regressive parameter estimation algorithm (RRPE) is proposed.

Fig. 2 illustrates the overview of the proposed method. SpBM consists of two stages: the model learning stage and the identification stage.

In the learning stage, the original observations are divided into some smaller non-overlapped sub-observations in a deterministic way. All the sub-observations sharing the same sub-pattern are collected to form the sub-pattern training sets. Afterwards, the style parameter and the content parameter, as well as their interaction matrix, are estimated from the sub-pattern training sets by a sub-pattern bilinear model learning algorithm.

In the identification stage, the style and the content are separated from each sub-pattern of the testing observation using the new ridge regressive parameter estimation algorithm. The obtained styles and contents are synthesized using majority voting and nearest neighbor algorithm to give the final decision of specific style and content.

### 3.1. *Sub-pattern bilinear model learning*

In SpBM, the original observations are divided into some smaller non-overlapped equally sized sub-observations in a deterministic way. And all the sub-observations which share the same specific sub-pattern are collected to build the corresponding training sub-pattern set. Fig. 3 shows the sub-pattern generating procedure described above. Then, the sub-pattern bilinear model learning algorithm is applied to each sub-pattern training set to estimate the model parameters. More specifically, assume the $K$ dimensional observation $\mathbf{z}^{sc}$ represents the whole pattern in style $s$ and content $c$, $s = 1,\ldots,T_s$, $c = 1,\ldots,T_c$. It is divided into $M$

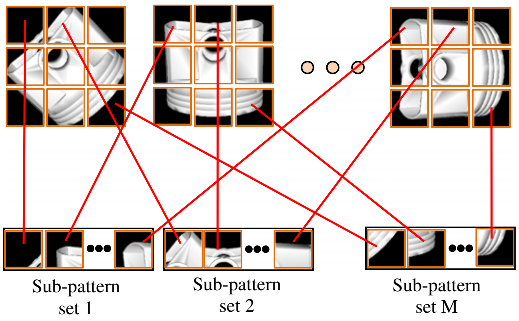


**Fig. 3.** Construction of sub-pattern sets.

sub-pattern observations of dimension $D$ ($K = M \times D$) as: $\mathbf{z}_m^{sc}$, $s = 1,\ldots,T_s$, $c = 1,\ldots,T_c$, $m = 1,\ldots,M$.

Similar to (1), each sub-observation can be represented by a linear combination of the style factor and the content factor as

$$\mathbf{z}_m^{sc} = \sum_{i=1}^{I}\sum_{j=1}^{J} \mathbf{w}_{mij}\mathbf{a}_{mi}^{s}\mathbf{b}_{mj}^{c}, \quad m = 1,\ldots,M \tag{14}$$

where $\mathbf{a}_m^s = (a_{m1}^s, a_{m2}^s, \ldots, a_{mI}^s)^{\mathrm{T}}$ and $\mathbf{b}_m^c = (b_{m1}^c, b_{m2}^c, \ldots, b_{mJ}^c)^{\mathrm{T}}$ are the style parameter and the content parameter of the $m$th sub-pattern, respectively, $\mathbf{w}_{mij}$ are a series of interaction vectors of the corresponding sub-pattern. Thus, the model parameters of each sub-pattern can be estimated by minimizing the following total square error:

$$E_m^{learn} = \sum_{s=1}^{T_s}\sum_{c=1}^{T_c} \left\| \mathbf{z}_m^{sc} - \sum_{i=1}^{I}\sum_{j=1}^{J} \mathbf{w}_{mij}\mathbf{a}_{mi}^{s}\mathbf{b}_{mj}^{c} \right\|^2, \quad m = 1,\ldots,M \tag{15}$$

The minimization of the total squared error of each sub-pattern can be solved by iterative singular value decomposition (SVD) algorithm similar to Algorithm 1. This gives the model parameters for each sub-pattern: $\{\mathbf{a}_m^s\}_{s=1}^{T_s}$, $\{\mathbf{b}_m^c\}_{c=1}^{T_c}$ and $\{\mathbf{w}_{mij}\}_{i=1,j=1}^{I,J}$, $m = 1,\ldots,M$.

**Algorithm 3.** Ridge regressive parameter estimation algorithm

```
(1) Initialize the content parameter b with the mean vector of B.
(2) Update the style parameter a by computing:
    $\mathbf{a} = (((\mathbf{W}\mathbf{b})^{\mathrm{VT}})^{\mathrm{T}}(\mathbf{W}\mathbf{b})^{\mathrm{VT}} + \lambda_1\mathbf{I})^{-1}((\mathbf{W}\mathbf{b})^{\mathrm{VT}})^{\mathrm{T}}\mathbf{z}^{test}$
(3) Update the content parameter b by computing:
    $\mathbf{b} = (((\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}})^{\mathrm{T}}(\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}} + \lambda_2\mathbf{I})^{-1}((\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}})^{\mathrm{T}}\mathbf{z}^{test}$
(4) Repeat steps (2) and (3) until the parameters a and b
    converge.
```

Thus, the local image information is preserved by means of styles and contents as well as their interaction vectors of each sub-pattern.

### 3.2. Ridge regressive parameter estimation

The parameter estimation algorithm given in Algorithm 2 may be instable due to the correlation of the data or noises [10]. To overcome this problem and stabilize the computation, a new ridge regressive parameter estimation algorithm is proposed in this subsection.

The parameter estimation algorithm of Algorithm 2 can be derived by minimizing the following objective function:

$$E(\mathbf{a},\mathbf{b}) = \sum_{k=1}^{K}\left(z_k - \sum_{i=1}^{I}\sum_{j=1}^{J} w_{ijk}a_i b_j\right)^2$$

$$= (\mathbf{z} - (\mathbf{W}\mathbf{b})^{\mathrm{VT}}\mathbf{a})^{\mathrm{T}}(\mathbf{z} - (\mathbf{W}\mathbf{b})^{\mathrm{VT}}\mathbf{a}) = (\mathbf{z} - (\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}}\mathbf{b})^{\mathrm{T}}(\mathbf{z} - (\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}}\mathbf{b}) \tag{16}$$

Then, the ridge regression is introduced into the parameter estimation by modifying the objective function as follows:

$$\begin{aligned} E(\mathbf{a},\mathbf{b}) &= \sum_{k=1}^{K}\left(z_k - \sum_{i=1}^{I}\sum_{j=1}^{J} w_{ijk}a_i b_j\right)^2 + \lambda_1 \sum_{i=1}^{I} a_i^2 + \lambda_2 \sum_{j=1}^{J} b_j^2 \\ &= (\mathbf{z} - (\mathbf{W}\mathbf{b})^{\mathrm{VT}}\mathbf{a})^{\mathrm{T}}(\mathbf{z} - (\mathbf{W}\mathbf{b})^{\mathrm{VT}}\mathbf{a}) + \lambda_1(\mathbf{a}^{\mathrm{T}}\mathbf{a}) + \lambda_2(\mathbf{b}^{\mathrm{T}}\mathbf{b}) \\ &= (\mathbf{z} - (\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}}\mathbf{b})^{\mathrm{T}}(\mathbf{z} - (\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}}\mathbf{b}) + \lambda_1(\mathbf{a}^{\mathrm{T}}\mathbf{a}) + \lambda_2(\mathbf{b}^{\mathrm{T}}\mathbf{b}) \end{aligned} \tag{17}$$

where $\lambda_1$ and $\lambda_2$ are positive constants. We differentiate the objective function $E(\mathbf{a},\mathbf{b})$ and set the partial derivatives equal to zeros: $\partial E/\partial \mathbf{a} = 0$ and $\partial E/\partial \mathbf{b} = 0$ to obtain:

$$\mathbf{a} = (((\mathbf{W}\mathbf{b})^{\mathrm{VT}})^{\mathrm{T}}(\mathbf{W}\mathbf{b})^{\mathrm{VT}} + \lambda_1\mathbf{I})^{-1}((\mathbf{W}\mathbf{b})^{\mathrm{VT}})^{\mathrm{T}}\mathbf{z} \tag{18}$$

$$\mathbf{b} = (((\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}})^{\mathrm{T}}(\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}} + \lambda_2\mathbf{I})^{-1}((\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}})^{\mathrm{T}}\mathbf{z} \tag{19}$$

Thus, the specific style and content of the testing observation can be estimated by iterating (18) and (19) until convergence. The details of the new ridge regressive parameter estimation algorithm are shown in Algorithm 3.

Therefore, the style and the content of each sub-pattern of a given testing observation $\mathbf{z}^{test}$ are estimated using Algorithm 3 instead of Algorithm 2.

By adding positive real numbers $\lambda_1$ and $\lambda_2$ to the diagonals of $((\mathbf{W}\mathbf{b})^{\mathrm{VT}})^{\mathrm{T}}(\mathbf{W}\mathbf{b})^{\mathrm{VT}}$ and $((\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}})^{\mathrm{T}}(\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}}$, respectively, ridge regression keeps (18) and (19) in well-conditioned (the condition numbers of (18) and (19) are constrained). Thus, the inverse operations in Algorithm 3 are stable and as a result the convergence of the model parameters is guaranteed. Typically, Algorithm 3 converges within 20 steps in our experiments. Theoretical analysis of the stability of proposed ridge regressive parameter estimation is given in the Appendix. The differences between RRPE and a similar algorithm proposed by Shin et al. [10] are also discussed in the Appendix.

### 3.3. Style and content identification

To finally determine the specific style and content of a given testing observation $\mathbf{z}^{test}$, a method which combines majority voting and minimal distance algorithm is proposed in this subsection. The style and the content of each sub-pattern, $\{\mathbf{a}_m\}_{m=1}^{M}$ and $\{\mathbf{b}_m\}_{m=1}^{M}$, are estimated by Algorithm 3 given in foregoing subsection.

Using (9) and (10), the distances between the estimated factor parameters and the model parameters are measured as

$$d_1(s,\mathbf{a}_m) = \|\mathbf{a}^s - \mathbf{a}_m\|, \quad s = 1,\ldots,T_s, \quad m = 1,\ldots,M \tag{20}$$

$$d_2(c,\mathbf{b}_m) = \|\mathbf{b}^c - \mathbf{b}_m\|, \quad c = 1,\ldots,T_c, \quad m = 1,\ldots,M \tag{21}$$

For each sub-pattern observation $\mathbf{z}_m^{test}$, the corresponding optimal style and content are computed as following:

$$s_m^{opt} = \arg_s \min d_1(s,\mathbf{a}_m), \quad m = 1,\ldots,M \tag{22}$$

$$c_m^{opt} = \arg_c \min d_2(c,\mathbf{b}_m), \quad m = 1,\ldots,M \tag{23}$$

Then, majority voting is used to find the best fitting style(s) and content(s) for the testing observation:

$$S^{opt} = \left\{ s \,\middle|\, \arg_s \max\left(\sum_{m=1}^{M} \delta_{s,m}^{\text{style}}\right) \right\} \tag{24}$$

$$C^{opt} = \left\{ c \,\middle|\, \arg_c \max\left(\sum_{m=1}^{M} \delta_{c,m}^{\text{content}}\right) \right\} \tag{25}$$

where

$$\delta_{s,m}^{style} = \begin{cases} 1, & s = s_m^{opt} \\ 0, & s \neq s_m^{opt} \end{cases}, \quad s = 1,\ldots,T_s, \quad m = 1,\ldots,M \tag{26}$$

$$\delta_{c,m}^{content} = \begin{cases} 1, & c = c_m^{opt} \\ 0, & c \neq c_m^{opt} \end{cases}, \quad c = 1,\ldots,T_c, \quad m = 1,\ldots,M \tag{27}$$

Since there may be more than one style or content identified, the final decision is given by computing the minimal distance between the estimated parameter and the learned model parameter

$$s^{Decision} = \arg_s \min(\min_m d_1(s,\mathbf{a}_m)), \quad s \in S^{opt} \tag{28}$$

$$c^{Decision} = \arg_c \min(\min_m d_2(c,\mathbf{b}_m)), \quad c \in C^{opt} \tag{29}$$

where $s^{Decision}$ and $c^{Decision}$ are the final identified style and content.

### 3.4. Time complexity discussion

The time complexity of SVD is $O(\min\{mn^2,m^2n\})$ for a $m \times n$ matrix [24]. Based on this, the time complexity of Algorithm 1 can be deduced: $O(QK(JT_s^2+IT_c^2))$, where $Q$ is the iteration steps. Therefore, the time complexity for SpBM training is $O\left(\sum_{i=1}^{M} Q_i (K/M)(JT_s^2+IT_c^2)\right)$, where $Q_i$ is the iteration steps of the $i$th sub-pattern.

The time complexity of Algorithms 2 and 3 are the same as: $O(Q(I^3+J^3))$. And the time complexity of parameter estimation of SpBM is $O\left(\sum_{i=1}^{M} Q_i(I^3+J^3)\right)$.

## 4. Experimental results and discussion

Bilinear models have been successfully applied to different kinds of tasks, including sparse coding of location and content in natural images [3], gait recognition [5,12], facial expression synthesis [7], facial recognition [10,11], 3D facial expression recognition [13], speaker adaption [14], spatio-temporal point distribution analysis [15], and so on.

In this paper, the proposed sub-pattern bilinear model is introduced into pose estimation of work-pieces to separate and estimate some key pose factors individually. The observation of the work-piece is affected by the following pose factors: the translations along the coordinate axis ($X$, $Y$, $Z$), the roll angle (rotation around $Z$-axis), the pitch angle (rotation around $Y$-axis) and the yaw angle (rotation around $X$-axis). Since translations can be well estimated by existing methods [16–21], we focus on separating and estimating the pitch angle and the yaw angle. The proposed method also can be extended to the separation and estimation of other factors.

Fig. 4 shows the illustrations of the pitch angle (denoted by $\varphi$) and the yaw angle (denoted by $\theta$) of a piston and a crank shaft, respectively.

Two kinds of data sets were used: simulated images and real images. For simulated images, the piston and the crank shaft were modeled in *SolidWorks* and then generated with an interval of 0.1° in 12 specific angle ranges (shown in Table 2) using *OPENGL*. That is, 12 specific data sets are generated for each work-piece. Each data set contains 400 images; and there are totally 4800 images for the piston and the crank shaft, respectively.

For the real images in our database, 840 images of the crank shaft were taken by the CCD camera with 704 × 576 pixels mounted on the industrial robot. The crank shaft was placed at 35 different yaw angles $\theta$ as: $\theta = -85°, -80°, \ldots, 80°, 85°$. For each yaw angle, the crank shaft was fixed at 24 different pitch angles as follows: $\varphi = 0°, 5°, 7.5°, 10°, \ldots, 57.5°, 60°$. All the images were clipped and resized into 240 × 240, containing the crank shaft located at the center position. Then, the crank shaft was segmented from the background manually. Fig. 5 shows some real images of the crank shaft in the database.

In order to evaluate the identification results with the same bias, each original data set is arranged into training set and testing set in the following way:

- Half of the images were sampled in an interlaced way along the *yaw* (*pitch*) direction to form the training set.
- The rest ones sampled in the similar interlaced way along the *yaw* (*pitch*) direction were used as testing set.

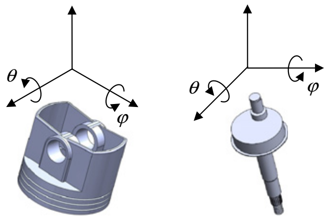


**Fig. 4.** Illustration of the pitch angle $\varphi$ and the yaw angle $\theta$ of a piston (left) and a crank shaft (right).

**Table 2**
Specific angle ranges of simulated images.

| Pitch $\varphi$ | Yaw $\theta$ | | |
|---|---|---|---|
| | 0° | 45° | 90° |
| 0° | $\theta \in [0°,2°)$<br>$\varphi \in [0°,2°)$ | $\theta \in [44°,46°)$<br>$\varphi \in [0°,2°)$ | $\theta \in [89°,91°)$<br>$\varphi \in [0°,2°)$ |
| 15° | $\theta \in [0°,2°)$<br>$\varphi \in [14°,16°)$ | $\theta \in [44°,46°)$<br>$\varphi \in [14°,16°)$ | $\theta \in [89°,91°)$<br>$\varphi \in [14°,16°)$ |
| 30° | $\theta \in [0°,2°)$<br>$\varphi \in [29°,31°)$ | $\theta \in [44°,46°)$<br>$\varphi \in [29°,31°)$ | $\theta \in [89°,91°)$<br>$\varphi \in [29°,31°)$ |
| 45° | $\theta \in [0°,2°)$<br>$\varphi \in [44°,46°)$ | $\theta \in [44°,46°)$<br>$\varphi \in [44°,46°)$ | $\theta \in [89°,91°)$<br>$\varphi \in [44°,46°)$ |

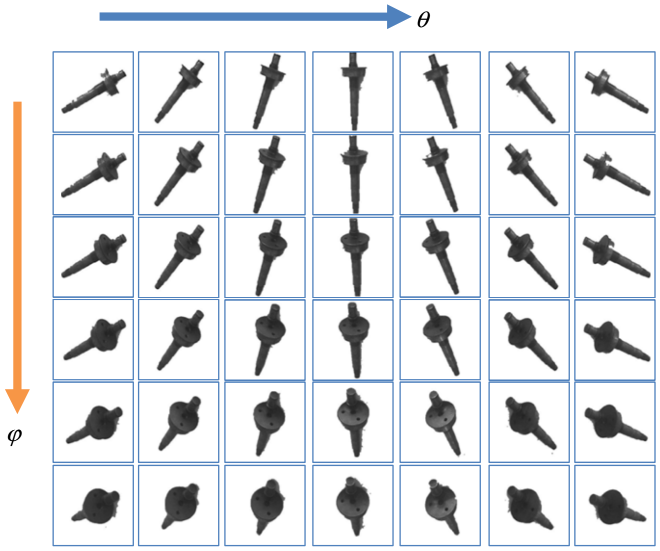


**Fig. 5.** Some real images of the crank shaft sampled from the real image data set. The images were captured in different poses with an interval of 2.5° along the pitch angle $\varphi$ and 5° along the yaw angle $\theta$.

To improve the calculation cost of the SpBM, the data sets were preprocessed using dimension reduction algorithm, such as random projection algorithm [22] in this paper.

### 4.1. Efficiency of ridge regressive parameter estimation (RRPE)

In the identification stage, we proposed a new ridge regressive parameter estimation algorithm to improve the performance of SpBM. In this subsection, the validation of RRPE was proven by comparing with other parameter estimation algorithms, such as the original parameter estimation algorithm (PE) [1] and Shin's algorithm (Shin's) [10].

One specific simulated image data set of the piston, in which the poses were in the range of $\varphi \in [29^\circ,31^\circ)$ and $\theta \in [44^\circ,46^\circ)$, was taken as the experimental data set.

The parameter $\lambda$ in Shin's algorithm [10] and the parameters $\lambda_1$ and $\lambda_2$ in RRPE were determined using a tuning process. 40 percent of the testing images were taken for evaluating the identification with respect to different values of each parameter: $\lambda$, $\lambda_1$ and $\lambda_2$. According the identification results, we chose the parameters that showed the best performance. To make the tuning process convenient, $\lambda_1$ and $\lambda_2$ were set the same value.

Firstly, we tested the performance of the proposed RRPE using testing data sets without noises. Experimental results are given in Table 3. All of the three pose estimation algorithms show satisfactory identification results. In the experiments, the parameters were set as $\lambda=1$ and $\lambda_1=\lambda_2=1$.

Secondly, we tested the performance of the proposed RRPE using testing data sets with salt and pepper noises whose density was 0.01. The optimal values of $\lambda$, $\lambda_1$ and $\lambda_2$ were determined first. As mentioned above, 40% of the testing images were taken for evaluation. Fig. 6 shows the identification results with different values of $\lambda$ in Shin's method. Fig. 7 shows the identification results with respect to different values of $\lambda_1$ and $\lambda_2$. According to Figs. 6 and 7, we chose the parameters as: $\lambda=100$ and $\lambda_1=\lambda_2=2000$.

Experimental results are given in Table 4. By introducing the ridge regression, both the Shin's algorithm and RRPE stabilized the matrix inverse computation in parameter estimation and consequently improved the identification results of SpBM. Compared to the original parameter estimation algorithm (PE) [1], Shin's algorithm increased the identification rate by about 5% while RRPE showed a more significant improvement of about 10%.

### 4.2. Performance of SpBM in non-occlusion cases

In this subsection, we check the performance of SpBM in the non-occlusion cases, i.e., the work-pieces are not occluded by other objects. The simulated images of the crank shaft and the piston were taken for evaluation.

**Table 3**
Comparison of identification results using testing data sets *without* noises (16 sub-patterns were used, i.e., $M=16$).

| Method | Error 0° | Error ≤ 0.1° |
|---|---|---|
| *Yaw angle identification rate* | | |
| SpBM+RRPE | **99.5%** | 100% |
| SpBM+Shin's | **99.5%** | 100% |
| SpBM+PE | 98.5% | 100% |
| *Pitch angle identification rate* | | |
| SpBM+RRPE | **97.5%** | 100% |
| SpBM+Shin's | 96.5% | 100% |
| SpBM+PE | 95.5% | 100% |

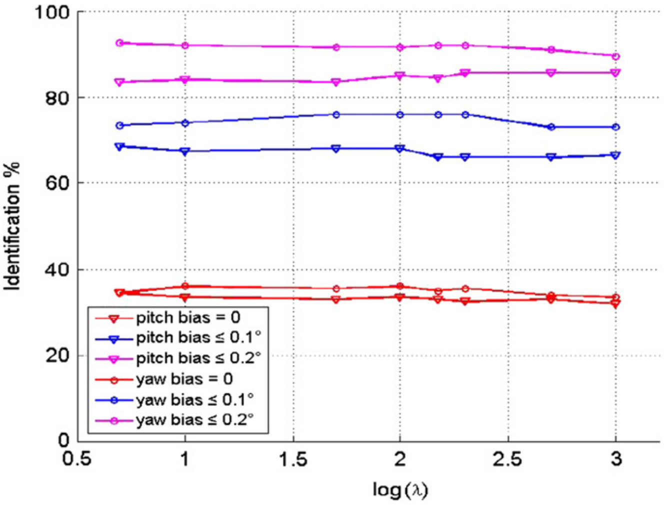


**Fig. 6.** Identification results with respect to different values of $\lambda$ in Shin's method.

**Table 4**
Comparison of identification results using testing data sets *with* salt and pepper noises (16 sub-patterns were used, i.e., $M=16$).

| Method | Error 0° | Error ≤ 0.1° | Error ≤ 0.2° |
|---|---|---|---|
| *Yaw angle identification rate* | | | |
| SpBM+RRPE | **45.5%** | **85%** | **94.5%** |
| SpBM+Shin's | 34% | 68% | 89% |
| SpBM+PE | 32% | 66% | 83% |
| *Pitch angle identification rate* | | | |
| SpBM+RRPE | **41.5%** | **87%** | **95%** |
| SpBM+Shin's | 36% | 74.5% | 88.5% |
| SpBM+PE | 29% | 69.5% | 85% |

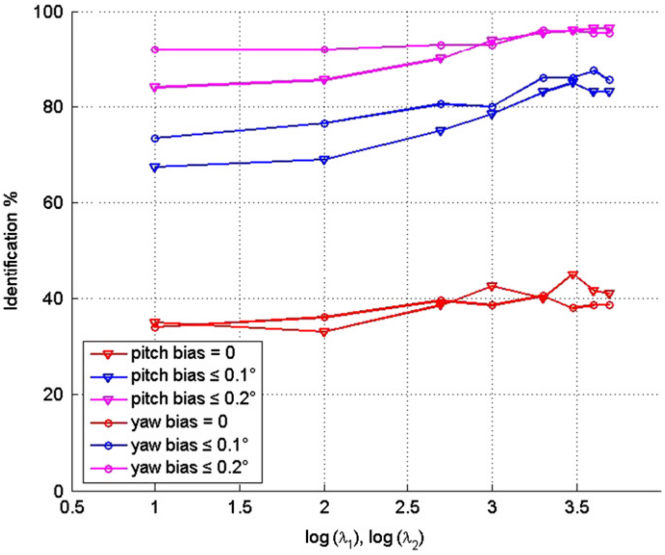


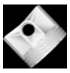
**Fig. 7.** Identification results with respect to different values of $\lambda_1$ and $\lambda_2$ in RRPE ($\lambda_1=\lambda_2$).

Experimental results are shown in Tables 5 and 6. For the crank shaft, 8 sub-patterns ($M=8$) were used when the yaw angle range was $\theta \in [0^\circ,2^\circ)$ or $\theta \in [89^\circ,91^\circ)$; 16 sub-patterns ($M=16$) were used when the yaw angle range was $\theta \in [44^\circ,46^\circ)$. All the yaw angles of the crank shaft could be identified correctly; and over 85% of the pitch yaws were identified correctly in most cases.

**Table 5**
Identification rate of the pitch and yaw angle of the crank shaft without occlusion (8 sub-patterns were used when $\theta \in [0°,2°)$ or $\theta \in [89°,91°)$; 16 sub-patterns were used when $\theta \in [44°,46°)$).

| Pitch $\varphi$ | Yaw $\theta$ | | |
|---|---|---|---|
| | 0° | 45° | 90° |
| 0° | $\theta$: 100%<br>$\varphi$: 93%<br> | $\theta$: 100%<br>$\varphi$: 89%<br> | $\theta$: 100%<br>$\varphi$: 93%<br> |
| 15° | $\theta$: 100%<br>$\varphi$: 86%<br> | $\theta$: 100%<br>$\varphi$: 82.5%<br> | $\theta$: 100%<br>$\varphi$: 84%<br> |
| 30° | $\theta$: 100%<br>$\varphi$: 93.5%<br> | $\theta$: 100%<br>$\varphi$: 86%<br> | $\theta$: 100%<br>$\varphi$: 87%<br> |
| 45° | $\theta$: 100%<br>$\varphi$: 93%<br> | $\theta$: 100%<br>$\varphi$: 98.5%<br> | $\theta$: 100%<br>$\varphi$: 97.5%<br> |

**Table 6**
Identification rate of the pitch and yaw angle of the piston (16 sub-patterns were used, i.e., $M = 16$).

| Pitch $\varphi$ | Yaw $\theta$ | | |
|---|---|---|---|
| | 0° | 45° | 90° |
| 0° | $\theta$: 94.5%<br>$\varphi$: 90%<br> | $\theta$: 98.5%<br>$\varphi$: 86%<br> | $\theta$: 94%<br>$\varphi$: 92%<br> |
| 15° | $\theta$: 100%<br>$\varphi$: 98%<br> | $\theta$: 99.5%<br>$\varphi$: 96.5%<br> | $\theta$: 100%<br>$\varphi$: 96%<br> |
| 30° | $\theta$:100%<br>$\varphi$: 98.5%<br> | $\theta$: 99%<br>$\varphi$: 99.5%<br> | $\theta$: 100%<br>$\varphi$: 100%<br> |
| 45° | $\theta$: 99%<br>$\varphi$: 99.5%<br> | $\theta$: 97%<br>$\varphi$: 100%<br> | $\theta$: 100%<br>$\varphi$: 99%<br> |

For the piston, 16 sub-patterns ($M = 16$) were used for identification. Over 95% of the pitch angle and the yaw angle are identified correctly in cases when the pitch angle is larger than 15°; and the identification rates are over 85% when the pitch angle is near 0°. SpBM shows a satisfactory result in non-occlusion cases.

Experimental results show that identification of the yaw angles is more accurate than the pitch angle identification. That is because the appearance of the objects changes more significantly as the yaw angle changes.

### *4.3. Performance of SpBM in occlusion cases*

Three different kinds of experiments were carried out to validate the performance of SpBM when the observations were with occlusions.

First, the identification consistency of SpBM in different pose situations is explored. Six sets of simulated piston images were selected and occlusions of the same size were added to these images at the same position. The six sets were organized into two groups: (1) the pitch angle range was $\varphi \in [14°,16°)$, with the yaw angles were in three different angle ranges: $\theta \in [0°,2°)$, $\theta \in [44°,46°)$ and$\theta \in [89°,91°)$; (2) the pitch angle range was $\varphi \in [44°,46°)$, with the yaw angles were in three different angle ranges: $\theta \in [0°,2°)$, $\theta \in [44°,46°)$ and$\theta \in [89°,91°)$.

We compared the identification results of existing bilinear models (BM) with SpBM. The global pattern was divided into 16 sub-patterns ($M = 16$). Experiment results are shown in Fig. 8. SpBM gave a good result in all the six simulated image sets even though some part of the piston was occluded. SpBM identified almost all the yaw angle correctly while no more than 90% of yaw angles were identified by BM. BM identified less than 20% of the pitch angles with occlusion correctly. While SpBM identified more than 95% of the pitch angles when the pitch angle range was $\varphi \in [14°,16°)$ and over 80% when $\varphi \in [44°,46°)$.

Second, the relationship between the identification performance of SpBM and the size of the occlusion was evaluated. We chose a specific simulated image set where the pose of the piston was in the following range: $\varphi \in [29°,31°)$, $\theta \in [44°,46°)$. The piston images in testing set were occluded with different portions from 10% to 50%.

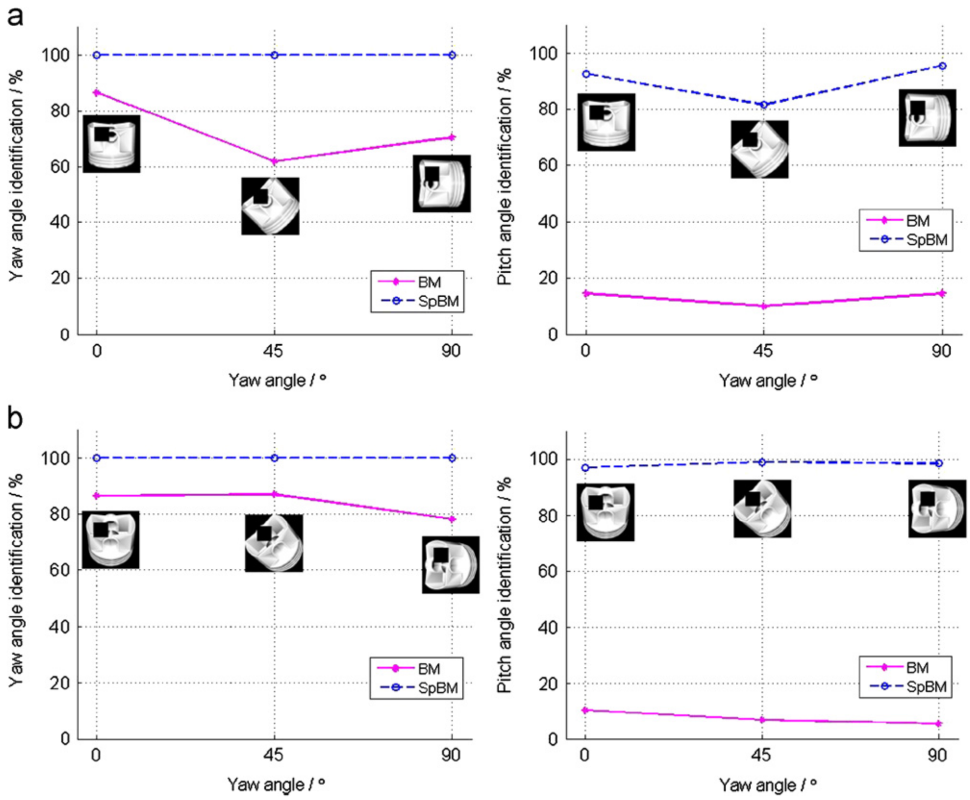


**Fig. 8.** The pitch and yaw angle identification of the piston with the same occlusion (16 sub-patterns were used, i.e., $M=16$). (a) Identification results of the piston with occlusion at pitch angle range of $\varphi \in [14^\circ,16^\circ)$ and three different yaw angle ranges: $\varphi \in [0^\circ,2^\circ)$ $\varphi \in [44^\circ,46^\circ)$ $\varphi \in [89^\circ,91^\circ)$ (Left: yaw angle; Right: pitch angle). (b) Identification results of the piston with occlusion at pitch angle range of $\varphi \in [44^\circ,46^\circ)$ and three different yaw angle ranges: $\varphi \in [0^\circ,2^\circ)$ $\varphi \in [44^\circ,46^\circ)$ $\varphi \in [89^\circ,91^\circ)$ (Left: yaw angle; Right: pitch angle).

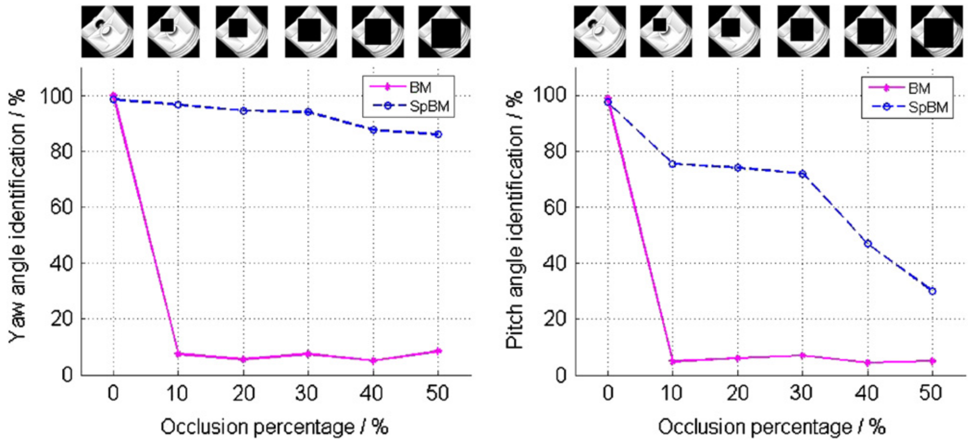


**Fig. 9.** Identification results of the piston at the specific angle range (pitch angle $\varphi \in [29^\circ,31^\circ)$, yaw angle $\theta \in [44^\circ,46^\circ)$ ) with different occlusion portions. (Left: yaw angle; Right: pitch angle. 16 sub-patterns were used $M=16$.)

As the experimental results shown in Fig. 9, the identification performance of SpBM decreased as the occluded portion increased. Yaw angle identification rate of SpBM kept up to 85% even though half of the piston was occluded. Identification rate of the pitch angle kept about 75% when 30% of the piston was occluded, but dropped rapidly when more portion of the piston was occluded. Meanwhile, the identification rate of BM dropped to less than 10% even when only 10% of the piston was occluded. Thus, for simulated images of work-pieces, SpBM is able to estimate the pitch angle and the yaw angle when no more than 30% of the work-piece is occluded.

Third, the identification performance of SpBM using real images was validated. Occlusions of the same size $85 \times 85$ were added to the testing crank shaft images $(240 \times 240)$ randomly. That is, the crank shaft in each testing image was occluded with different portion in different positions. Fig. 10 shows some real images of the crank shaft with random occlusion.

Both SpBM and BM were used to identify the pitch angle and the yaw angle. Fig. 11 shows the experimental results. SpBM showed a great improvement in yaw angle identification that over 80% testing images are within an error criterion of 5°. Pitch angle identification is improved by increasing the identification rate within an error criterion of 2.5°.

The parameters $\lambda_1$ and $\lambda_2$ were sensitive to different identification conditions, for instance, different work-pieces, different noise conditions or different occlusion sizes. However, when the pitch angle and the yaw angle were identified under the same condition, the values of $\lambda_1$ and $\lambda_2$ can be set the same values. For example, $\lambda_1$ and $\lambda_2$ were set the same value as $\lambda_1=\lambda_2=5$ in the six experiments in which the pitch angle and the yaw angle were identified from the simulated piston images with occlusions of the same size (Fig. 8). In contrast, $\lambda_1$ and $\lambda_2$ were set different values when different portions of the work-piece were occluded (Fig. 9).

### 4.4. Comparison between SpBM and LCM

Lots of works have been presented for pose estimation [30–32]. Especially, the work of Ullman and Basri [32], in which the objects was represented by combining the object models in a linear way, shows a similar mechanism in representing the object with new pose. Therefore, in this subsection, we compared the pose estimation performance of the Sub-pattern Bilinear Model (SpBM) with Ullman's approach, e.g. the Linear Combination of Models (LCM [32]).

Eighty-one simulated images of piston, in which the poses are in the range of $\varphi\in[29.6^\circ,30.4^\circ]$ and $\theta\in[44.6^\circ,45.4^\circ]$ with intervals of 0.1°, were taken for testing the performances of SpBM and LCM. The testing images were evaluated using another 49 simulated images of piston whose pose ranges were in $\varphi\in[27^\circ,33^\circ]$ and $\theta\in[42^\circ,48^\circ]$ with intervals of 1°.

The two approaches were compared in non-occlusion cases and occlusion cases. In occlusion experiments, black blocks of size $30\times30$ were added to the testing images ($240\times240$) randomly. For LCM, six template images were used and corresponding features were extracted by Harris interest point detector. Fig. 12 shows one of the synthesized images of the piston using LCM. For SpBM, the parameters $\lambda_1$ and $\lambda_2$ were set as $\lambda_1=\lambda_2=5$. The experimental results of non-occlusion cases and occlusion cases are shown in Tables 7 and 8, respectively. The results show that SpBM outperforms LCM in both non-occlusion cases and occlusion cases within an error criterion of 1°. But both the two approaches could identify all the poses correctly within an error criterion of 2°. Since LCM models the object by combining the corresponding visible points on the object surface from six models in a linear way, LCM shows robust performance in occlusion cases. However, it seems that LCM fails to distinguish poses of high precision such as 0.1° because of the deviation of features extraction and the rounding error of coordinate computed by linear combination.

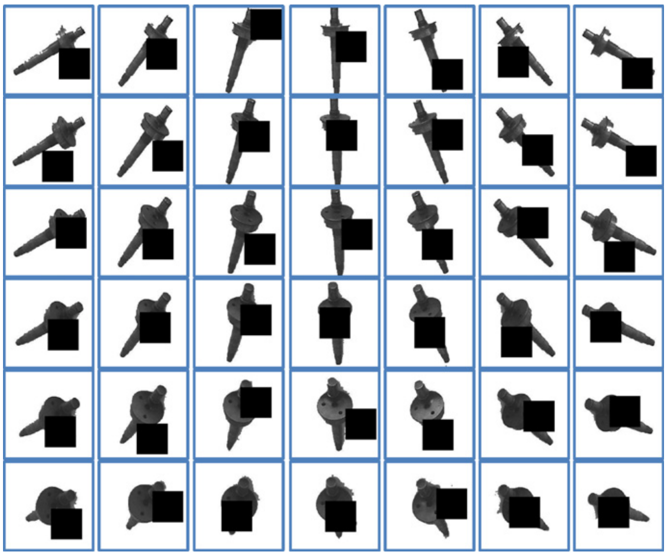


Fig. 10. Some testing real images sampled from the data set of the crank shaft with occlusion.

## 5. Conclusions

In this paper, a novel sub-pattern bilinear model (SpBM) which constructs the sub-pattern bilinear model utilizing local patterns instead of global patterns is proposed. The specific style and content of testing observations are identified by synthesizing the discriminative information provided by the local sub-patterns. Thus, SpBM is able to obtain good identification results in the cases when local image information cannot be obtained caused by occlusions or image noises. To further improve the identification performance of SpBM, we presented a new ridge regressive parameter estimation algorithm (RRPE) which introduces the ridge regression into the parameter estimation algorithm by modifying the objective function. RRPE stabilizes the matrix inverse computation and guarantees the convergence of parameter estimation. The theoretical analysis of RRPE is also given to prove its efficiency. Finally, the proposed sub-pattern bilinear model is introduced into pose estimation of work-pieces to

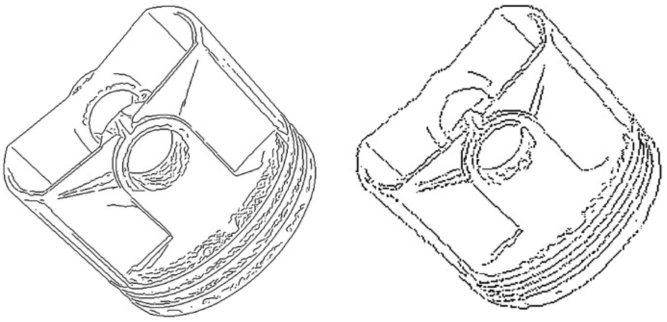


Fig. 12. The left one is one of the edge images of the piston; and the right one is its corresponding synthesized image using LCM.

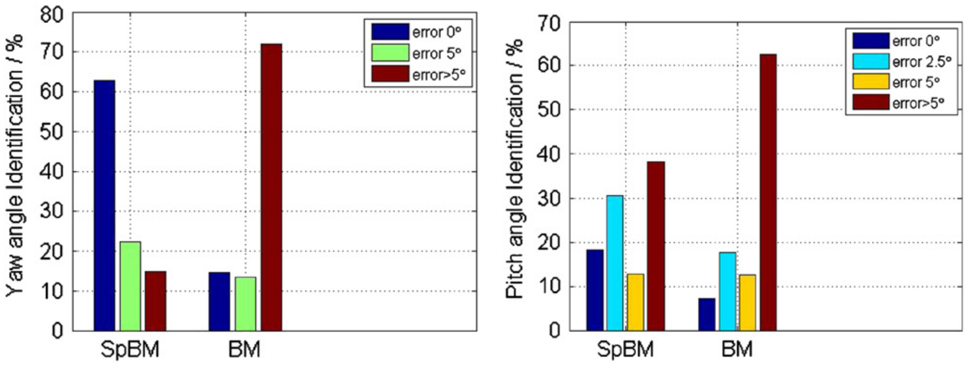


Fig. 11. Identification results of the crank shaft with occlusion using real image data set. (Left: yaw angle; Right: pitch angle. 16 sub-patterns were used, i.e., $M=16$.)

**Table 7**
Comparison of pose estimations between SpBM and LCM in non-occlusion cases.

| Method | Error ≤ 1° | Error ≤ 2° |
|---|---|---|
| *Yaw angle identification rate* | | |
| SpBM | **100%** | 100% |
| LCM | 65.4% | 100% |
| *Pitch angle identification rate* | | |
| SpBM | **87.7%** | 100% |
| LCM | 66.7% | 100% |

**Table 8**
Comparison of pose estimations between SpBM and LCM in occlusion cases.

| Method | Error ≤ 1° | Error ≤ 2° |
|---|---|---|
| *Yaw angle identification rate* | | |
| SpBM | **97.5%** | 100% |
| LCM | 65.4% | 100% |
| *Pitch angle identification rate* | | |
| SpBM | **74.1%** | 100% |
| LCM | 64.2% | 100% |

separate and estimate some key pose factors individually. Experimental results demonstrate the robust performance in the estimation of the pitch angle and the yaw angle.

## Acknowledgments

The work was supported in part by the National Natural Science Foundation (NNSF) of China under Grants 61033011, 61100098, 61101221 and 61105085, and by the Outstanding Youth Fund of the NNSF of China under Grant 60725310.

## Appendix

In Algorithm 3, one factor parameters are estimated by fixing the other factor parameters. Therefore, (18) and (19) can be interpreted as the following traditional ridge regressions, respectively:

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \left( \sum_{k=1}^{K} \left( z_k - \sum_{i=1}^{I} x_{a,ik} a_i \right)^2 + \lambda_1 \sum_{i=1}^{I} a_i^2 + C_a \right)$$
$$= \arg\min_{\mathbf{a}} ((\mathbf{z} - \mathbf{X}_a \mathbf{a})^{\mathrm{T}} (\mathbf{z} - \mathbf{X}_a \mathbf{a}) + \lambda_1 \mathbf{a}^{\mathrm{T}} \mathbf{a} + C_a), \tag{A.1}$$

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} \left( \sum_{k=1}^{K} \left( z_k - \sum_{j=1}^{J} x_{b,jk} b_j \right)^2 + \lambda_2 \sum_{j=1}^{J} b_j^2 + C_b \right)$$
$$= \arg\min_{\mathbf{b}} ((\mathbf{z} - \mathbf{X}_b \mathbf{b})^{\mathrm{T}} (\mathbf{z} - \mathbf{X}_b \mathbf{b}) + \lambda_2 \mathbf{b}^{\mathrm{T}} \mathbf{b} + C_b), \tag{A.2}$$

where $\mathbf{X}_a = (\mathbf{Wb})^{\mathrm{VT}}$, $\mathbf{X}_b = (\mathbf{W}^{\mathrm{VT}}\mathbf{a})^{\mathrm{VT}}$, $C_a = \lambda_2 \mathbf{b}^{\mathrm{T}}\mathbf{b}$, $C_b = \lambda_1 \mathbf{a}^{\mathrm{T}}\mathbf{a}$.

Then, the variances of the parameter vectors $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ in Algorithm 3 are computed as

$$\mathrm{Var}(\hat{\mathbf{a}}) = E\left[ (\hat{\mathbf{a}} - E(\hat{\mathbf{a}}))(\hat{\mathbf{a}} - E(\hat{\mathbf{a}}))^{\mathrm{T}} \right]$$
$$= E\Big\{ ((\mathbf{X}_a^{\mathrm{T}}\mathbf{X}_a + \lambda_1 \mathbf{I})^{-1} \mathbf{X}_a^{\mathrm{T}} \mathbf{z} - (\mathbf{X}_a^{\mathrm{T}}\mathbf{X}_a + \lambda_1 \mathbf{I})^{-1} \mathbf{X}_a^{\mathrm{T}} \mathbf{X}_a \mathbf{a})$$
$$\times ((\mathbf{X}_a^{\mathrm{T}}\mathbf{X}_a + \lambda_1 \mathbf{I})^{-1} \mathbf{X}_a^{\mathrm{T}} \mathbf{z} - (\mathbf{X}_a^{\mathrm{T}}\mathbf{X}_a + \lambda_1 \mathbf{I})^{-1} \mathbf{X}_a^{\mathrm{T}} \mathbf{X}_a \mathbf{a})^{\mathrm{T}} \Big\}$$
$$= E(((\mathbf{X}_a^{\mathrm{T}}\mathbf{X}_a + \lambda_1 \mathbf{I})^{-1} \mathbf{X}_a^{\mathrm{T}} (\mathbf{z} - \mathbf{X}_a \mathbf{a}))((\mathbf{X}_a^{\mathrm{T}}\mathbf{X}_a + \lambda_1 \mathbf{I})^{-1} \mathbf{X}_a^{\mathrm{T}} (\mathbf{z} - \mathbf{X}_a \mathbf{a}))^{\mathrm{T}})$$
$$= E((\mathbf{X}_a^{\mathrm{T}}\mathbf{X}_a + \lambda_1 \mathbf{I})^{-1} \mathbf{X}_a^{\mathrm{T}} \mathbf{e}\mathbf{e}^{\mathrm{T}} \mathbf{X}_a (\mathbf{X}_a^{\mathrm{T}}\mathbf{X}_a + \lambda_1 \mathbf{I})^{-1})$$
$$= \sigma \sum_{i=1}^{I} \mathbf{v}_{a,i} \left( \frac{d_{a,i}}{(d_{a,i} + \lambda_1)^2} \right) \mathbf{v}_{a,i}^{\mathrm{T}} \tag{A.3}$$

$$\mathrm{Var}(\hat{\mathbf{b}}) = E\left[ (\hat{\mathbf{b}} - E(\hat{\mathbf{b}}))(\hat{\mathbf{b}} - E(\hat{\mathbf{b}}))^{\mathrm{T}} \right]$$
$$= \sigma \sum_{j=1}^{J} \mathbf{v}_{b,j} \left( \frac{d_{b,j}}{(d_{b,j} + \lambda_2)^2} \right) \mathbf{v}_{b,j}^{\mathrm{T}} \tag{A.4}$$

where $\mathbf{e}$ is an error vector that is assumed to be an independent and identically distributed random variable with mean 0 and variance $\sigma^2$; $\mathbf{X}_a$ and $\mathbf{X}_b$ are dependent variable matrices of $\mathbf{a}$ and $\mathbf{b}$, respectively; $\mathbf{X}_a = \mathbf{U}_a \mathbf{D}_a \mathbf{V}_a^{\mathrm{T}}$, $\mathbf{v}_{a,i}$ is the $i$th column of $\mathbf{V}_a$ and $d_{a,i}$ is an eigenvalue corresponding to $\mathbf{v}_{a,i}$; $\mathbf{X}_b = \mathbf{U}_b \mathbf{D}_b \mathbf{V}_b^{\mathrm{T}}$, $\mathbf{v}_{b,j}$ is the $j$th column of $\mathbf{V}_b$ and $d_{b,j}$ is an eigenvalue corresponding to $\mathbf{v}_{b,j}$.

Thus, by adding the positive values $\lambda_1$ and $\lambda_2$ to the diagonal terms of $\mathbf{X}_a^{\mathrm{T}}\mathbf{X}_a$ and $\mathbf{X}_b^{\mathrm{T}}\mathbf{X}_b$, the proposed method reduces the variance of the parameters $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ to stabilize the regression model.

In [10], Shin et al. proposed similar ridge regressive pose estimation by revised the objective function as follows:

$$E(\mathbf{a},\mathbf{b}) = \sum_{k=1}^{K} \left( z_k - \sum_{i=1}^{I} \sum_{j=1}^{J} w_{ijk} a_i b_j \right)^2 + \lambda \sum_{i=1}^{I} \sum_{j=1}^{J} (a_i b_j)^2 \tag{A.5}$$

In their algorithm, the variants of the two estimated parameters are:

$$\mathrm{Var}(\hat{\mathbf{a}}) = \sigma \sum_{i=1}^{I} \mathbf{v}_{a,i} \left( \frac{d_{a,i}}{\left( d_{a,i} + \lambda \mathbf{b}^{\mathrm{T}} \mathbf{b} \right)^2} \right) \mathbf{v}_{a,i}^{\mathrm{T}} \tag{A.6}$$

$$\mathrm{Var}\left(\hat{\mathbf{b}}\right) = \sigma \sum_{j=1}^{J} \mathbf{v}_{b,j} \left( \frac{d_{b,j}}{(d_{b,j} + \lambda \mathbf{a}^{\mathrm{T}} \mathbf{a})^2} \right) \mathbf{v}_{b,j}^{\mathrm{T}} \tag{A.7}$$

The differences between RRPE and Shin's algorithm lie in following aspects:

- Different penalty factors are proposed. $\lambda \mathbf{a}^{\mathrm{T}} \mathbf{a} \mathbf{b}^{\mathrm{T}} \mathbf{b}$ is chosen in Shin's algorithm while $\lambda_1(\mathbf{a}^{\mathrm{T}}\mathbf{a}) + \lambda_2(\mathbf{b}^{\mathrm{T}}\mathbf{b})$ is given in this paper.
- Different positive numbers are added to the denominator of the variances of estimated parameters. $\lambda \mathbf{b}^{\mathrm{T}}\mathbf{b}$ and $\lambda \mathbf{a}^{\mathrm{T}}\mathbf{a}$ are added in Shin's algorithm while $\lambda_1$ and $\lambda_2$ are added in our algorithm.
- The ridge regression proposed by Shin might be instability in two extreme cases: (1) $\mathbf{a}^{\mathrm{T}}\mathbf{a} \to 0$and $d_{b,j} \to 0$; (2) $\mathbf{b}^{\mathrm{T}}\mathbf{b} \to 0$ and $d_{a,i} \to 0$. The proposed ridge regressive algorithm keeps stable when $\lambda_1 > 0$ and $\lambda_2 > 0$.

## References

[1] J.B. Tenenbaum, W.T. Freeman, Separating style and content with bilinear models, Neural Comput 12 (6) (2000) 1247–1283.

[2] X.B. Gao, C.N. Tian, Multi-view face recognition based on tensor subspace analysis and view manifold modeling, Neurocomputing 72 (16–18) (2009) 3742–3750.

[3] D.B. Grimes, R.P.N. Rao, Bilinear sparse coding for invariant vision, Neural Comput 17 (1) (2005) 47–73.

[4] M. Alex, O. Vasilescu, Demetri Terzopoulos, Multilinear analysis of image ensembles: TensorFaces, in: Proceedings of the European Conference on Computer Vision, vol. 1, 2002, pp. 447–460.

[5] A. Elgammal, C.S. Lee, Separating style and content on a nonlinear manifold, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. 478–485.

[6] Y. Du, X. Lin, Nonlinear factorization models using kernel approaches, in: Proceedings of the Asian Conference on Computer Vision, vol. 1, 2004, pp. 426–431.

[7] C. Zhou, X.Y. Lin, Facial expressional image synthesis controlled by emotional parameters, Pattern Recognition Lett. 26 (16) (2005) 2611–2627.

[8] J. Gonzalez-Moraet al., Bilinear active appearance models, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[9] D.H. Marimont, B.A. Wandell, Linear models of surface and illuminant spectra, J. Opt. Soc. Am. A 9 (11) (1992) 1905–1913.

[10] D. Shin, H.S. Lee, D. Kim, Illumination-robust face recognition using ridge regressive bilinear models, Pattern Recognition Lett. 29 (1) (2008) 49–58.

[11] H. Yan, J. Yang, J. Yang, Bimode model for face recognition and face representation, Neurocomputing 74 (5) (2011) 741–748.

[12] F. Cuzzolin, Using bilinear models for view-invariant action and identity recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 1701–1708.

[13] I. Mpiperis, S. Malassiotis, M.G. Strintzis, Bilinear models for 3-D face and facial expression recognition, IEEE Trans. Inform. Forensic Security 3 (3) (2008) 498–511.

[14] H.J. Song, H.S. Kim, Bilinear model-based maximum likelihood linear regression speaker adaptation framework, Signal Process. Lett. 16 (12) (2009) 1063–1066.

[15] C. Hoogendoorn, F.M. Sukno, S. Ordás, A.F. Frangi, Bilinear models for spatio-temporal point distribution analysis, application to extrapolation of left ventricular, biventricular and whole heart cardiac dynamics, Int. J. Comput. Vision 85 (3) (2009) 237–252.

[16] G. Borgefors, Hierarchical chamfer matching: a parametric edge matching algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 10 (6) (1988) 849–865.

[17] D.F. Dementhon, L.S. Davis, Model-based object pose in 25 lines of code, Int. J. Comput. Vision 15 (1–2) (1995) 123–141.

[18] H. Murase, S.K. Nayar, Visual learning and recognition of 3-D objects from appearance, Int. J. Comput. Vision 14 (1) (1995) 5–24.

[19] A. Ansar, K. Daniilidis, Linear pose estimation from points or lines, IEEE Trans. Pattern Anal. Mach. Intell. 25 (5) (2003) 578–589.

[20] 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, May 2010, pp. 2028–2035.

[21] D. Breitenreicher, C. Schnörr, Model-based multiple rigid object detection and registration in unstructured range data, Int. J. Comput. Vision 92 (1) (2011) 32–52.

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

[23] J.R. Magnus, H. Neudecker, Matrix Differential Calculus With Applications in Statistics and Econometrics, Wiley, New York, 1988.

[24] 〈http://campar.in.tum.de/twiki/pub/Chair/TeachingWs05ComputerVision/3DCV_svd_000.pdf〉.

[25] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, John Wiley & Sons, 2001.

[26] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Networks 13 (4–5) (2000) 411–430.

[27] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, Face recognition by independent component analysis, IEEE Trans. Neural Networks 13 (6) (2002).

[28] J.H. Lee, et al., Speech feature extraction using independent component analysis, ICASSP 3 (2000) 1631–1634.

[29] A.M. Bronstein1, et al., Sparse ICA for blind separation of transmitted and reflected images, Int. J. Imaging Syst. Technol. 15 (1) (2005) 84–91.

[30] J.J. Koenderink, A.J. van Doorn, The internal representation of solid shape with respect to vision, Biol. Cybern. 32 (4) (1979) 211–216.

[31] S.J. Dickinson, A.P. Pentland, A. Rosenfeld, 3-D shape recovery using distributed aspect matching, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 174–198.

[32] S. Ullman, R. Basri, Recognition by linear combinations of models, IEEE Trans. Pattern Anal. Mach. Intell. 13 (10) (1991) 992–1006.

**Zhicai Ou** received the B.Eng. degree in automation from University of Science and Technology of China, Heifei, China, in 2006. He is currently working toward the Ph.D. degree in the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interest covers robotic vision, manufacturing automation, robotic grasping and assembly.

**Peng Wang** received the B.Eng. degree in electrical engineering and automation from Harbin Engineering University, Harbin, China, in 2004, the M.Eng. degree in automation science and engineering from the Harbin Institute of Technology, Harbin, in 2007, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010.

He is currently an Assistant Professor with the Institute of Automation, Chinese Academy of Sciences. He has worked on computer vision, image processing, and intelligent robots. His current research interests include computer vision, human–robot interaction, high precision sensing and control, and visual perception.

**Jianhua Su** received the B.Eng. degree in electronic and information engineering from Beijing Jiaotong University, Beijing, China, in 1999, the M.Eng. degree in Electronic and information engineering from the Beijing Jiaotong University, Beijing, in 2004, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2009.

He is currently an Assistant Professor with the Institute of Automation, Chinese Academy of Sciences. His background is in the fields of control theory, robotics, automation, and manufacturing. His current research interests include intelligent robot system and train control system.

**Hong Qiao** received the B.Eng. degree in hydraulics and control and the M.Eng. degree in robotics from Xian Jiaotong University, Xian, China, the M.Phil. degree in robotics control from the Industrial Control Center, University of Strathclyde, Strathclyde, UK, and the Ph.D. degree in robotics and artificial intelligence from De Montfort University, Leicester, UK in 1995.

She was a University Research Fellow with De Montfort University from 1995 to 1997. She was a Research Assistant Professor from 1997 to 2000 and an Assistant Professor from 2000 to 2002 with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong. Since January 2002, she has been a Lecturer with the School of Informatics, University of Manchester, Manchester, UK. Currently, she is also a Professor with the Laboratory of Complex Systems and Intelligent Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China. She first proposed the concept of "the attractive region in strategy investigation," which has successfully been applied by herself in robot assembly, robot grasping, and part recognition. The work has been reported in Advanced Manufacturing Alert (Wiley, 1999). Her current research interests include information-based strategy investigation, robotics and intelligent agents, animation, machine learning (neural networks and support vector machines), and pattern recognition.

Dr. Qiao is a member of the Program Committee of the IEEE International Conference on Robotics and Automation from 2001 to 2004. She is currently the Associate Editors of the IEEE Transaction on Systems, Man, and Cybernetics—Part B and IEEE Transaction on Automation Science and Engineering.