Sparse Bayesian Regression for Head Pose Estimation

Yong Ma, Yoshinori Konishi, Koichi Kinoshita, Shihong Lao, Masato Kawade
Sensing & Control Lab., Omron Corporation, Kyoto, Japan, 619-0283
Email: *ma@ari.ncl.omron.co.jp*

Abstract

This paper presents a high performance head pose estimation system based on the newly-proposed sparse Bayesian regression technique (Relevance Vector Machine, RVM) and sparse representation of facial patterns. In our system, after localizing 20 key facial points, sparse features of these points are extracted to represent facial property, and then RVM is utilized to learn the relation between the sparse representation and yaw and pitch angle. Because RVM requires only a very few kernel functions, it can guarantee better generalization, faster speed and less memory in a practical implementation. To thoroughly evaluate the performance of our system, we compare it with conventional methods such as CCA, Kernel CCA, SVR on a large database; In experiments, we also investigate the influence of the facial points localization error on pose estimation by using manually labelled results and automatically localized results separately, and the influence of different features on pose estimation such as geometrical features and texture features. These experimental results demonstrate that our system can estimate face pose more accurately, robustly and fast than those based on conventional methods.

1. Introduction

Like face detection [1] and facial points localization [2], head pose estimation also plays a very important role in vision-based automatic face perception processes. Most of pose-invariant face recognition systems [3] need to normalize face patterns according to their poses prior to extracting features. Other man-machine interaction applications, such as gaze tracking system [5] and driver mental status monitoring system [4], also need to estimate head orientations before making decisions.

Till now, a variety of methods have been proposed to deal with this problem. These methods can be grossly divided into three types. The first type is the classification-based methods which use different learning methods, for example, tree-structured AdaBoost pose classifier [8], ANN [9] and soft margin AdaBoost [12], to train classifiers to classify the 2D face holistic appearance into a specific pose class. So this type of methods only can estimate coarse pose ranges instead of precise yaw (rotation around the neck, from left profile to right profile) and pitch (rotation up and down) angle values. The second type uses regression algorithms such as CCA [6], Kernel CCA [7], SVR [10] to estimate head pose precisely based on 2D holistic appearance or sparse representation. Because these regression models are not sparse enough, they generally need a large run-time library to do regression. For example, Moon's SVR-based method [10] selects 35,151 support vectors for pitch and yaw estimation. So these methods are not suitable for embedded devices and real-time processes. The last type uses 3D face model, such as texture-mapped cylinder face model [11][19], to estimate head pose. Although the convergence speed of these methods can be accelerated greatly by motion tracking, they are heavily influenced by the estimation precision of the initial frames and the estimation error would be accumulated easily. Besides these, this type of methods can not deal with occlusion and large non-rigid motion effectively.

Different from above methods, in this paper, we will combine the state-of-arts sparse Bayesian regression techniques (Relevance Vector Machines, RVM) [15][16] with sparse representation of facial images to achieve both high accuracy and high feasibility. Till now, RVM has been successfully applied to problems such as 3D human pose estimation [17], visual tracking [18]. In our system, we first localize 20 key facial points, and then extract features of these facial points. RVM regression algorithm is used to learn the relation between this sparse representation and head pose. Experimental results on a large database demonstrate the effectiveness of our method. And several important investigations, such as the influence of different facial sparse representation methods and localization precision of facial points on pose estimation, are conducted.

The rest of the paper is organized as follows. In section 2, we introduce the theoretic framework of sparse Bayesian framework. In section 3, the details of our head pose estimation system are described. In section 4, thorough comparisons are made among different head pose estimation methods on a large scale face database. Finally, section 5 gives the conclusions.

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

2. Sparse statistical regression under Bayesian framework

The Relevance Vector Machine, or RVM, was proposed by Tipping [15] as a Bayesian treatment of the sparse learning problem. The RVM preserves the generalization properties of the SVM, yet it yields a more sparse output. Different from SVM's structural risk minimization principle utilized to control its sparsity, RVM introduces more powerful Gaussian priors on each weight parameter and each weight parameter is controlled by its own individual scale hyper-parameter. In the following parts, we will explain the RVM and its training for regression problems detailedly.

2.1. Relevance Vector Machine

For a regression problem, given a training dataset $\{(\mathbf{x}_i, t_i) \mid i = 1,\ldots,N\}$, the following generalized linear regression model can be used to describe the mapping relation between the input pattern vector $\mathbf{x}$ and the scalar target t:

$$t = y(\mathbf{x},\mathbf{w}) + \varepsilon \; .$$

And $y(\mathbf{x},\mathbf{w})$ can be expressed as a linearly weighted sum of some kernel basis functions $K(\mathbf{x}_i,\mathbf{x})$:

$$y(\mathbf{x},\mathbf{w}) = \sum_{i=1}^{N} w_i K(\mathbf{x}_i,\mathbf{x}) + w_0 \, ,$$

Where $\varepsilon \sim N(0,\sigma^2)$ and $\mathbf{w} = [w_0,\ldots,w_N]$. Here w_i is the weight parameter associated with one basis function.

So the conditional distribution is also a Gaussian distribution.

$$p(t \mid \mathbf{x},\mathbf{w},\sigma^2) = N(y(\mathbf{x},\mathbf{w}),\sigma^2)$$

Because all the training samples are assumed to be independent, the likelihood of the training dataset can be written as

$$p(\mathbf{t} \mid \{\mathbf{x}_1,\ldots,\mathbf{x}_N\},\mathbf{w},\sigma^2) = N(\mathbf{\Phi}\mathbf{w},\sigma^2)$$

Where $\mathbf{\Phi}$ is the $N \times (N+1)$ design matrix and contains the intra-training set kernel values:

$$\mathbf{\Phi}_{ij} = K(\mathbf{x}_i,\mathbf{x}_{j-1}) \text{ and } \mathbf{\Phi}_{i1} = 1 .$$

Maximum-likelihood estimation of $\mathbf{w}$ and σ^2 from the above equation will generally lead to overfitting problem. So a prior preference for smoother functions over the weights is specified:

$$p(\mathbf{w} \mid \boldsymbol{\alpha}) = \prod_{i=0}^{N} N(w_i \mid 0, \alpha_i^{-1})$$

Considering this hyperprior, the posterior over the weights can be obtained from Bayes's rule [15]:

$$p(\mathbf{w} \mid \mathbf{t},\boldsymbol{\alpha},\sigma^2) = N(\boldsymbol{\mu},\mathbf{\Sigma}) \qquad (1)$$

with

$$\Sigma^{-1} = \sigma^{-2}\Phi^T\Phi + diag(\boldsymbol{\alpha})$$

$$\boldsymbol{\mu} = \sigma^{-2}\Sigma\Phi^T\mathbf{t}$$

2.2. Training algorithm of RVM

According to (1), $\boldsymbol{\mu}$ and Σ are determined by the hyperparmeters $\boldsymbol{\alpha}$, σ^2 which characterize different models and training involves finding these values via Bayesian model selection. To do this, the *marginal likelihood*, $p(t \mid \alpha,\sigma^2)$ is maximized using a variant of gradient ascent. The training algorithm proposed in [15] begins optimizing over the entire training set, and the examples are pruned whenever the associated hyper-parameters fall below a threshold, leading to the final sparse solution. Those examples left are termed as relevance vectors. This RVM training algorithm takes a lot time for large training sets because in each iteration the posterior (1) needs to be evaluated which involves matrix inversion. In our system, we adopt another fast marginal likelihood maximization algorithm proposed in [16]. It divides all training samples into three types analytically similar to SMO [23] strategy, begins with only one basis vector and increases/reduces basis vectors one by one during iterations until the convergence of posterior has been attained.

The kernel function used here is the Gaussian kernel function:

$$K(\mathbf{x}_i,\mathbf{x}) = \exp(-\gamma \left\|\mathbf{x}-\mathbf{x}_i\right\|^2) .$$

The width parameter γ has an important effect on the outcome of the training: too small and the RVM will overfit, too large and it will underfit (large σ^2 and poor accuracy at runtime). We set this value here by experiment.

3. Robust head pose estimation based on sparse Bayesian regression

According to the requirements of real applications, the acceptable face pose range in our system is [-60, 60] degrees for yaw and [-30, 30] degrees for pitch, as shown in figure 1. The rotation of head in image plane (roll) can be estimated easily, for example according to the positions of two eyecenters. So it is not considered here.

The overview of our system is given in figure 2. First we use a multi-view face detector proposed in [20] to detect faces and then use the method proposed in [21] to localize 20 key facial points based on texture appearances. These 20 points are the most obvious feature points in face area such as eye inner corner, eye outer corner, eye center, mouth corners and their interpolated positions(figure 2); Then the scale and direction (roll rotation)

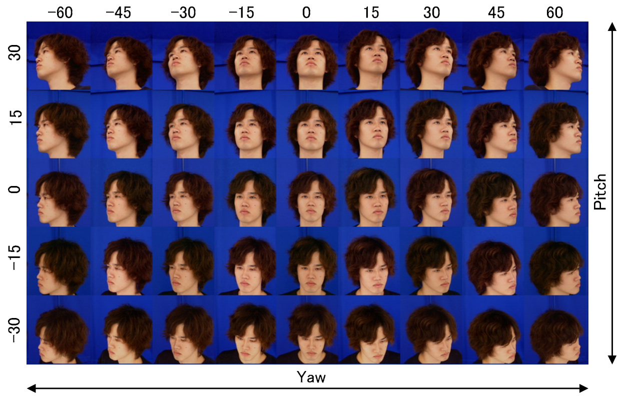

Figure 1 Head pose estimation range

of the face area can be normalized according to two eye-centers before training and testing; Lastly two types of feature are extracted according to these facial points. The relation between the sparse representations and head pose (yaw and pitch) can be learned using the above RVM regression training method separately.

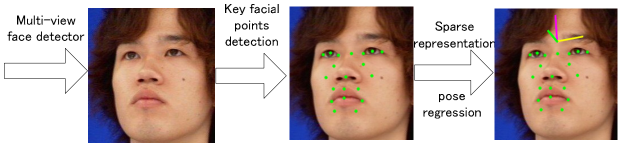

Figure 2 Flowchart of the pose estimation system

3.1. Two sparse representation of facial patterns

Here we adopt two different feature sets to represent the face patterns.

The first kind of sparse representation is the normalized coordinates of 20 key facial points using the following method. Assuming the positions of 20 key facial points in image coordinate are (x_i, y_i) $i = 1,\ldots,20$, the gravity center is (x_c, y_c):

$$x_c = \tfrac{1}{20}\sum_{i=1}^{20} x_i\,,\quad y_c = \tfrac{1}{20}\sum_{i=1}^{20} y_i$$

$$ad = \tfrac{1}{20}\sum_{i=1}^{20}((x_i - x_c)^2 + (y_i - y_c)^2)^{1/2}$$

Then the normalized position of a facial point is

$$x_i^{norm} = \frac{((x_i - x_c)^2 + (y_i - y_c)^2)^{1/2}}{ad}(x_i - x_c)$$

$$y_i^{norm} = \frac{((x_i - x_c)^2 + (y_i - y_c)^2)^{1/2}}{ad}(y_i - y_c)$$

After concatenating these normalized positions, a 40 dimension feature vector can be got. The yaw and pitch can be estimated from this geometrical feature separately.

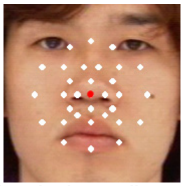

Figure 3 A retinal sampling grid example

The second feature set is extracted using simplified Gabor wavelet [22] and retinal sampling [22] techniques. Retinal sampling, which sets sampling points densely towards the center and sparsely away from the center, can extract local features and global features in a well-balanced manner. For every key facial point, the retinal sampling grid is shown in figure 3. And for every sampling point, 4 orientations and 1 scale of wavelet features are extracted. So totally $4 \times 32 \times 20 = 2560$ dimension feature vector can be extracted for a face. The yaw and pitch can be estimated from this texture feature separately.

4. Experiments

4.1. Experiments setup

The training and testing experiments are conducted on Softpia* database. As shown in figure 1, faces in Softpia database are divided into 45 classes according to their pitch and yaw angles with the interval of 15 degrees. Every class contains 250 faces labelled with 20 key facial points manually. Among them, 80 faces are selected randomly for training and 170 for testing. So totally 3600 samples are used for training and 7650 are for testing.

The evaluation protocol used in our experiments is as followed. Assuming that the ground truth pitch and yaw angles of a face are $\beta_{yaw}^{GT}, \beta_{pitch}^{GT}$, and the estimated pitch and yaw are $\beta_{yaw}^{est}\ \beta_{pitch}^{est}$, if the following condition is satisfied, the estimation result is considered correct:

$$\left|\beta_{yaw}^{est} - \beta_{yaw}^{GT}\right| \le Th \ \text{ and } \left|\beta_{pitch}^{est} - \beta_{pitch}^{GT}\right| \le Th$$

Here Th is the threshold. In our experiments, we set the threshold as 7.5 and 10 degrees.

4.2. Experimental result

In our experiments, RVM regression method is compared with traditional regression methods such as CCA, kernel CCA and SVR on above geometrical feature (GF) and texture feature (TF). And in order to compare their robustness, the same experiments are conducted based on

* The facial data in this paper are used by permission of Softpia Japan, Research and Development Division, HOIP Laboratory. It is strictly prohibited to copy, use, or distribute the facial data without permission.

manually labelled 20 facial points (GT) and automatic localized 20 facial points (AU) separately.

In the experiments, when CCA algorithm [6] is implemented on TF, we first use PCA to reduce the dimension from 2560 to 512 to deal with singularity of scatter matrix; When kernel CCA [7] is implemented, Gaussian kernel is used and regularization parameter [6][7] is 0.01 for GF and 0.1 for TF; And in SVR implementation, $\varepsilon-$SVR regression [24] framework and Gaussian kernel are adopted.

The experimental result is given in table 1. From the table we can see that compared with CCA, linear RVM needs 1/2 of features to achieve similar or even better performance, and compared with kernel CCA and Gaussian SVR, Gaussian RVM only needs 1/15~1/6 of total training samples to achieve similar performance. So the run-time speed of RVM is much faster than that of kernel CCA and it is more feasible for embedded devices.

From the table, we also can conclude that the system based on texture feature is more robust to localization errors than that based on geometrical feature; For example the performance of Gaussian RVM using TF is similar to that of Gaussian RVM using GF based on GT result, but Gaussian RVM using TF based on AU is much better than Gaussian RVM using GF. So are kernel CCA and Gaussian SVR. The reason for this is that texture feature using Gabor transform contains more redundant information than geometrical feature. It matches the conclusion of [2] very well that the face recognition system based on Gabor wavelet feature was very robust to facial points localization error.

5. Conclusion

In this paper, a robust automatic head pose estimation system is presented. Thorough experiments on a large test set compared with conventional methods demonstrate its effectiveness.

References

[1] C.Huang, H.Ai, etc., "Boosting nested cascade detector for multi-view face detection", ICPR, 2004.

[2] Y.Ma, Z.Wang, etc. "Robust facial point localization under probabilistic framework",IEEE AFGR, 2004, Korea.

[3] A. Pentland, B. Moghaddam etc., "View-based and modular Eigenspaces for face recognition", CVPR 1994.

[4] X. Liu, Y. Zhu, K. Fujimura, "Real time pose classification for driver monitoring". IEEE Conf. on ITS, 2002,174-178

[5] Q. Liu, Y. Rui, etc., "Automating camera management for lecture room environment". In: ACM CHI, 2000.

[6] T.Melzer etc.,"Appearance models based on kernel canonical correlation analysis",PR,2003,36:1961-1971

[7] H.Murase, S.K.Nayar, "Visual learning and recognition of 3-d objects from appearance", International Journal of Computer Vision 14 (1) 5-24, 1995.

[8] Z. Yang, H.Ai, etc. "Multi-view face pose classification by tree-structured classifier", ICIP 2005, II 358-361.

[9] S.Baluja, M.Sahami, etc., "Efficient face orientation discrimination", ICIP 2004, 589-592.

[10] H. Moon and M. Miller, "Estimating facial pose from a sparse representation", ICIP2004.

[11] M.LaCascia,etc."Fast, reliable head tracking under varying illumination:An approach based on registration of textured-mapped3Dmodels".IEEE trans.PAMI,22(4):322–336, 2000.

[12] Y.Guo, G.Poulton, etc. "Soft margin AdaBoost for face pose classification", IEEE conf. on ASSP, 2003.

[13] L.Morency, P.Sundberg, etc. "Pose estimation using 3D view-based Eigensapces", ICCV workshop on Analysis and Modeling of Face and Gesture, pp. 45-52, October 2003.

[14] V.Blanz and T.Vetter, "Face recognition based on fitting a 3d morphable model". IEEE. PAMI, 25:1063-1074, 2003

[15] M.Tipping, "The relevance vector machines". Advances in NIPs vol.12, pp. 652–658, 1999

[16] M.Tipping,A.Faul. "Fast marginal likelihood maximization for sparse Bayesian models". Proc. of the 9th Int. Workshop on Artificial Intelligence and Statistics, 2003

[17] A.Agarwal and B.Triggs. "3D human pose from silhouettes by Relevance Vector Regression". IEEE CVPR, 2004.

[18] O.Williams,A.Blake,"Sparse Bayesian learning for efficient visual tracking". IEEE PAMI, vol.27, No.8, Aug. 2005.

[19] J. Xiao, T. Moriyama, etc. "Robust full-motion recovery of head by dynamic templates and re-registration techniques". Int. J. Imaging Syst. & Tech.,Vol.13, Sep.,2003, pp.85-94.

[20] Y.Yamshita, S.Lao, etc. "A fast Omni-directional face detection system", ICCV2005, Demo.

[21] K.Kinoshita,Y. Ma, etc. "Fast and robust facial points localization and pose estimation system", ICCV2005, Demo

[22] F.Smeraldi, J.Bigun. "Facial feature detection by saccadic exploration of the Gabor decomposition", ICIP, 1998.

[23] J.Platt. "Sequential minimal optimization: a fast algorithm for training Support Vector Machines", MSR-TR-98-14.

[24] T.Joachims, "Making large-Scale SVM Learning Practical". Advances in Kernel Methods, MIT-Press, 1999

Table 1 Pose estimation correct rates of different methods on different features

Method/Threshold		CCA		Kernel CCA		Gaussian SVR		Linear RVM		Gaussian RVM	
		7.5	10	7.5	10	7.5	10	7.5	10	7.5	10
GF	GT	66.0%	82.7%	88.5%	95.8%	87.0%	95.7%	66.2%	82.5%	88.5%	96.3%
	AU	61.7%	80.1%	72.2%	85.0%	77.6%	89.1%	61.3%	79.2%	74.3%	87.2%
	No of *basis vector*s or features*	80		7200		2774		42		498	
TF	GT	80.8%	93.1%	86.7%	95.5%	88.2%	95.9%	83.0%	94.1%	88.6%	96.7%
	AU	72.2%	86.8%	80.0%	91.7%	81.5%	91.4%	74.4%	88.6%	81.7%	91.9%
	No. of *basis vector*s or features*	1024		7200		2974		769		1279	

*Please note here that for linear classifiers (CCA and Linear RVM), all or part of features are selected to construct decision functions, and for nonlinear classifiers (Kernel CCA, Gaussian SVR and Gaussian RVM), all or part of training samples are selected to construct decision functions (here we call these selected training samples as *basis vectors* instead of Support Vectors or Relevance Vectors separately).