Applying Discriminatively Optimized Feature Transform for HMM-based Off-line Handwriting Recognition

Jin Chen[*]
Lehigh University
Bethlehem, PA 18015
jic207@cse.lehigh.edu

Bing Zhang
Raytheon BBN Technologies
Cambridge, MA 02138
bzhang@bbn.com

Huaigu Cao
Raytheon BBN Technologies
Cambridge, MA 02138
hcao@bbn.com

Rohit Prasad
Raytheon BBN Technologies
Cambridge, MA 02138
rprasad@bbn.com

Prem Natarajan
Raytheon BBN Technologies
Cambridge, MA 02138
pnatarajan@bbn.com

Abstract

Feature extraction is an important step in off-line handwriting recognition systems to represent raw handwriting in a low-dimensional, tractable feature space. Traditionally, linear feature transforms such as Principle Component Analysis (PCA), Linear Discriminative Analysis (LDA) are commonly used. The assumptions they make, however, usually cannot be satisfied in practice and thus the best performance is not obtained. In this paper, we apply the Region-Dependent non-linear feature Transform (RDT) to handwriting recognition. RDT is one type of non-linear feature transforms which captures the discriminating power much better than traditional linear ones. We justify the effectiveness of RDT on handwriting features using an HMM-based handwriting recognition system on an Arabic handwriting dataset, which consists of 38K pages of handwriting, over 3M handwritten words. Experimental results show that RDT is able to decrease the word error rates (WERs) relatively by 4% to 7% with statistical significance, comparing to two LDA-based baseline systems.

1. Introduction

Handwriting recognition is a task of converting the handwriting signal, an ink sequence or a static text image, into a text transcription. Due to the complexity of the raw signal, handwriting recognition systems usually work on discrete features extracted from the raw signal. In addition, feature transform is employed for the following two reasons: (1) to resolve the correlation between dimensions of features. (2) to alleviate the data sparsity problem in a high dimensional space by forcing discriminating power into fewer dimensions.

The commonly used linear feature transforms are Principle Component Analysis [6] (PCA), Linear Discriminative Analysis [6] (LDA), and Heteroscedastic Linear Discriminant analysis (HLDA) [11]. Although conceptually and/or computationally intuitive, these linear feature transforms do not always produce the optimal transform for practical problems. For example, PCA does not preserve any discriminative information between classes. From this perspective, LDA is more suitable for compressing information within each class, while preserving distances between different classes for multi-class classification problems, e.g., handwriting recognition. However, both LDA and PCA may fail when the classes are not normally distributed. In addition, LDA will also fail when the within-class distributions are *heteroscedastic* [10]. HLDA relaxes the assumption of equal-variance data and modifies the objection function such that discriminative information is forced into the first several dimensions [11]. However, HLDA is more sensitive than LDA to correlated data and thus performs poorly when feature frame concate-

[*] Jin Chen performed this work during his summer internship at Raytheon BBN Technologies.

[1] This paper is based upon work supported by the DARPA MADCAT Program.

[2] The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.182

nation is commonly used in HMM-based handwriting recognition systems.

In this paper, we employ a non-linear feature transform called Region Dependent Transform [22] (RDT), which has proven to be effective in automatic speech recognition. RDT is discriminatively optimized using the criterion called Minimum Phoneme Errors [18] (MPE), which relates the transform directly to the *Word Error Rates* (WERs) in recognition systems.

In the remaining of this paper, we will first briefly describe the idea of RDT in Section 2 and its training in Section 3. Next, we will explain the main modules in the HMM-based handwriting recognition engine in Section 4. Then in Section 5, we introduce the experimental setup including the data preparation, preprocessing, and the features we use for evaluation. Finally we present experimental results in Section 6 and conclude in Section 7.

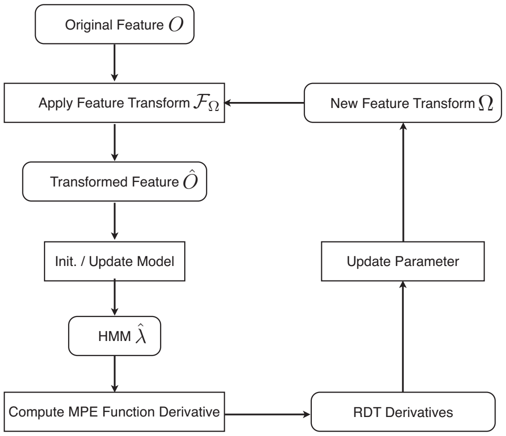
Original Feature O
Apply Feature Transform $\mathcal{F}_\Omega$
New Feature Transform Ω
Transformed Feature $\hat{O}$
Init. / Update Model
Update Parameter
HMM $\hat{\lambda}$
Compute MPE Function Derivative
RDT Derivatives

Figure 1. A flowchart of RDT parameter optimization under the HMM-based handwriting recognition model [21].

2. Region Dependent Transform [22]

RDT is a region-dependent transform by which the feature space is divided into N regions, and each region has a different transform to apply. The feature space is divided by a global Gaussian mixture model (GMM) on all training samples. Quoting the definitions in [21], RDT can be written as a collection of N vector-to-vector functions:

$$\mathcal{F}_{RDT}(o_t) = \sum_{i=1}^{N} k_t^{(i)} f_i(o_t) \qquad (1)$$

where o_t is a given input observation vector at time t, usually a frame-concatenated feature vector with hundreds of dimensions. $f_i(\cdot)$ is one of a collection of transformation functions: $\{f_i : \mathbf{R}^n \mapsto \mathbf{R}^p \mid i \in [1, N]\}$, and $k_t^{(i)}$ is the posterior probability of region i given the observation vector, and the total likelihood with respect to all Gaussians, assuming equal prior probability of each region:

$$k_t^{(i)} = \frac{\mathcal{N}(\hat{o}_t \mid \mu_i, \sigma_i)}{\sum_{j=1}^{N} \mathcal{N}(\hat{o}_t \mid \mu_j, \sigma_j)} \qquad (2)$$

where $\mathcal{N}(\cdot)$ is the probability density function of a multi-variant Gaussian, and $\hat{o}_t = A o_t$ is the projected feature vector computed from the projection matrix A. The addition in Eq. 1 is to compensate ambiguity of assigning a feature vector to one single region, i.e., several regions can overlap in the Gaussian mixture model.

Ideally, the region posterior $k_t^{(i)}$ should be computed using o_t. But since features are usually concatenated from the neighbor sliding windows, GMM's covariance matrices acquired in the original feature space will not be diagonal due to high correlations. Thus, we approximate the likelihood in a lower dimensional space using a boost-strapping feature transform A (usually LDA).

In this work, we chose a linear transform in each region for simplicity:

$$f_i(o_t) = A_i o_t \qquad (3)$$

The subscript i in A_i indicates that the projection matrix may differ in different regions, although at the initial stage, all A_i are the same if we use a global LDA transform to project all original feature vectors.

It should be noted that RDT is not the only way to employ non-linear feature transforms, however, several alternative ways are shown to exist under the same framework of RDT [17, 5].

3. RDT Training

The workflow for the RDT training is shown in Figure 1. As we can see, this is an iterative framework for RDT parameter estimation. First, we apply an initial feature transform to original feature vectors so that the following RDT training is conducted in the projected feature space. Next, we use the single-pass retraining (SPR) [8] to initialize the HMM model λ. Then we compute the derivatives of the MPE objective functions with respect to the feature transform parameter vector Ω:

$$\Omega = [\text{vec}(A_1)^T, \text{vec}(A_2)^T, \ldots, \text{vec}(A_N)^T] \qquad (4)$$

where “vec$(\cdot)$” is a function that returns a column vector from a matrix by stacking the column vectors of the matrix. After computing the “direct-RDT” and “indirect-RDT” derivatives (explained below), we obtain the updating rules for the RDT parameter Ω that is needed to update the feature transform functions.

RDT training tries to find a set of region dependent feature transforms such that the resulting HMMs has the best MPE score if it is trained under the ML criterion using the transformed features. In [21], the MPE objective function [16, 18] can be formulated as follows:

$$\mathcal{H}(\mathbf{X}, \lambda) = \sum_{r=1}^{R} \sum_{k=1}^{K(r)} \frac{P(X_r \mid W_{rk}, \lambda)^{\beta} P(W_{rk}) \alpha(W_{rk})}{\sum_{k'=1}^{K(r)} P(X_r \mid W_{rk'}, \lambda)^{\beta} P(W_{rk'})} \quad (5)$$

where:

i) $\mathbf{X} = \{X_1, \ldots, X_R\}$ is the sequence of all transformed feature vectors of all R training text lines.

ii) λ is a set of trainable parameters of the HMM model for handwriting recognition, which consists of the means and covariances of the Gaussian components. This is also referred as the *codebook* in the literature.

iii) $X_r = \{x_1^r, \ldots, x_{T(r)}^r\}$ is the sequence of transformed feature vectors of text line r, whose length is $T(r)$.

iv) W_{rk} is one of the $K(r)$ hypotheses for the word sequence of text line r.

v) $\alpha(W_{rk})$ is the character accuracy score of the hypothesis against the reference transcription.

vi) β is a constant used to adjust the dynamic range of the scoring.

Two types of dependence exist between the objective function $\mathcal{H}(\mathbf{X}, \lambda)$ and the feature transform $f_i(\cdot)$. First, the transformed feature vector x_t is the output of the feature transform function $f_i(\cdot)$. Second, the means and covariances in the HMM model also depend on the feature transform since they are updated by the Maximum Likelihood (ML) criterion during RDT training.

Now we compute the MPE function derivatives with respect to the transform parameter Ω using the chain rule:

$$\frac{\partial \mathcal{H}(\mathbf{X}, \lambda)}{\partial \Omega_k} = \sum_{r=1}^{R} \sum_{t=1}^{T(r)} \frac{\partial \mathcal{H}(\mathbf{X}, \lambda)}{\partial x_t^r} \frac{\partial x_t^r}{\partial \Omega_k} \quad (6)$$

Note that the second term $\frac{\partial x_t^r}{\partial \Omega_k}$ only depends on the feature transform so it is easy to compute. For the first term, we can further rewrite it as:

$$\frac{\partial \mathcal{H}(\mathbf{X}, \lambda)}{\partial x_t^r} = \left(\frac{\partial \mathcal{H}(\mathbf{X}, \lambda)}{\partial x_t^r} \right)_{\lambda} + \left(\frac{\partial \mathcal{H}(\mathbf{X}, \lambda)}{\partial \lambda} \right)_{x_t^r} \frac{\partial \lambda}{\partial x_t^r} \quad (7)$$

where the first term is referred as the “direct-RDT” which assumes the HMM model λ holds constant, so $\mathcal{H}(\cdot, \cdot)$ only depends on the transformed features x_t^r. The second term is referred as the “indirect-RDT” since the means and covariances in λ also depend on the transformed feature vectors x_t^r. For further details on deducing the two derivatives into computational forms, refer to [21].

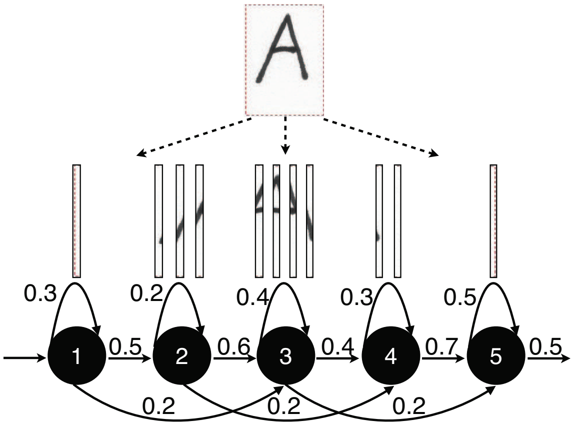

Figure 2. An example of a 5-state HMM with the Bakis topology. To recognize a character, the HMM model aligns the features to the internal states and evaluate the entire observation probability.

4. HMM-based Handwriting Recognition System

Figure 2 shows an HMM model that is used for handwritten text line recognition. This is a character HMM that can be easily extended to word HMMs and text line HMMs, by concatenating character ones.

4.1 HMM Training

HMM training estimates parameters for the HMM model: transition probabilities between states, the weights of Gaussian mixture components, and the means and diagonal covariances of Gaussian components in the output probability distribution. In our experiments, we use 14-state Bakis HMMs for handwritten text line recognition [14]. The training is done by the Baum-Welch algorithm [15].

In addition, character tied method (CTM) [9] and State tied method (STM) [12] are employed in the HMM training for extensive parameter estimation. CTM means that all characters share the same codebook, *i.e.*, the same means and diagonal covariances of the Gaussian components, and they only differ in the weights c_k. In STM, each state of all HMMs associated with every character has one codebook.

4.2 HMM Recognition

The HMM recognition, also known as *decoding*, refers to the process of determining an optimal state sequence through the HMM network that has the maximum probability. The decoding is done by the two-pass (forward and backward) beam search [1, 19]. In addition, we use a trigram language model to guide the path search in which the probabilities of all triplets of words in the training dataset are computed.

5. Experimental Setup

5.1 Data Preparation

The dataset is an Arabic handwritten document corpus that includes multiple writers, various paper materials and writing instruments. For each page, every handwritten text line and its associate words are labeled by polygons. For each polygon, the text transcription is also provided.

We employed two types of pre-processing to facilitate handwriting recognition. First, we detected and removed pre-printed rulings, meanwhile any broken strokes were recovered by stroke generation [2]. Second, we made use of the text line polygons to crop out handwritten text lines for HMM training and recognition. Then, to minimize the variations of handwriting, we corrected the slant of each word to make sure the vertical strokes are perpendicular to the baseline [20]. Finally, we divided handwritten text lines into three groups: training, developing, and testing sets. A breakdown of these three datasets is shown in Table 1.

Table 1. Datasets in experiments.

	Training	Developing	Testing
# of pages	37,608	560	545
# of lines	658,691	9,029	10,017
# of words	3,820,433	56,912	61,023

5.2 Feature Extraction

5.2.1 Gabor Features

We applied Gabor filtering in four directions and we used the magnitude as the response for feature extraction [3]. After the filtering, we divided the input image frame equally into 12 rows and computed the feature vector as a concatenation of features in each grid. This resulted in 48-D Gabor features.

5.2.2 GSC Features

Gradient-Structural-Concavity (GSC) features are multi-resolution features that combine three different shape attributes of the text: gradient representing the local orientation of strokes; structure information that extends the gradient to longer distance and provides information about stroke trajectories; and concavity that captures stroke relationships at long distances [7]. We divided each image frame equally into 12 rows and the feature vectors are 96-D features.

5.2.3 Other Features

Other features include Percentiles (20-D), Angle (6-D), Correlation (6-D), and Energy (1-D). Details on these features are explained in [13].

5.2.4 Feature Transform

As we can see, for each image frame, we have an 177-D feature vector. Then, features in adjacent frames are concatenated so that the long-span feature can express more structure information. This process results in 531-D feature vectors.

After computing all different types of features for all training samples, we normalized feature vectors at all their dimensions. This is to ensure that for each dimension, all the training samples are approximately normally distributed $\mathcal{N}(0, \mathbf{I})$. Next, we applied the LDA transform to reduce feature dimensions to 17 for the following experiments.

5.3 Baseline Systems

The first baseline system uses LDA for feature transform and the ML criterion for HMM training. The second baseline uses LDA for feature transform and the MPE criterion for HMM training. The proposed systems use RDT for feature transforms.

6. Experimental Results

6.1 RDT Training

First we show the iterative training for RDT in Figure 3. Since we used the MPE objective function for RDT training and MPE correlates well with WERs, we

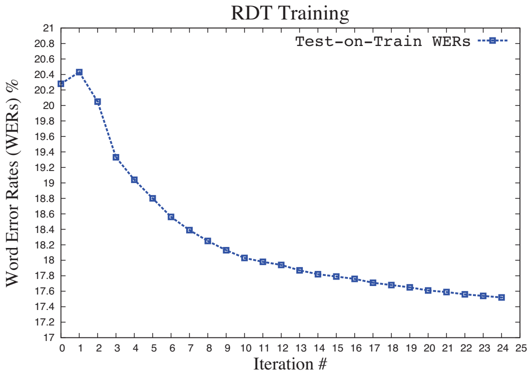

Figure 3. A diagram showing the RDT training process.

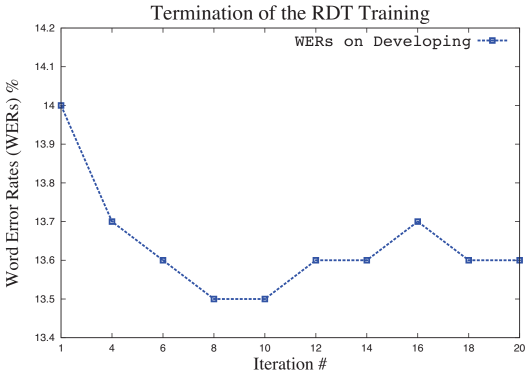

Figure 4. A diagram showing performance on the validation dataset.

can see that with more iterations, WERs gradually decreased. In addition, at the end of each iteration, we obtained an updated HMM model and updated feature transforms. Of course, we should bear in mind that the lowest WER on the training dataset does not guarantee the best performance on the testing dataset. Indeed, we should also take into account of the *over-fitting* issue.

Ideally, termination of the RDT training should be decided by a regression test on the developing dataset. Due to the scale of the datasets, however, we chose to try a subset of all training products (*i.e.*, codebooks, feature transforms) on the developing dataset and select the one that minimizes WERs. A performance diagram of RDT on the developing dataset is shown in Figure 4. At the initial stage, the RDT training iteratively optimizes the HMM codebook and the feature transform in each region. When it arrived at a local minimum (Iteration 10), the training started to over-fit. This is why the WER curve descends in the RDT training (Figure 3), while starting to ascend after Iteration 10 in Figure 4.

6.2 HMM Recognition using Optimized Features

After RDT training, we first applied the optimized feature transforms to the developing and the testing samples. Then, we replaced the codebooks in the baseline systems with the one from RDT training.

We used Developing Dataset to estimate the weights for two scores: *acoustic* score from the HMM decoding and *alignment* score from the language model scoring. Then we used the optimized weights to compute the WERs on Testing Dataset. The WERs on Developing (best only) and Testing are shown in Table 2. As we can see, MPE-based systems outperform ML-based systems. In addition, although the ML-based systems have low WERs already, the RDT-boosted systems are still able to gain in performance, 7% relative for the ML system and 4% relative for the MPE system.

Table 2. System performance comparison, shown in word error rates (WERs).

	Developing	**Testing**
LDA+ML	14.3%	12.8%
RDT+ML	13.5%	**11.9%**
LDA+MPE	12.2%	10.7%
RDT+MPE	11.6%	**10.3%**

6.3 Statistical Significance Test

We now justify that these performance gains are statistically significant. Dietterich [4] suggests evaluating the difference between two classification approaches using the McNemar's test:

$$Z^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{10} + n_{01}}. \tag{8}$$

where we first divided misclassified samples into two groups, and then stated the hypothesis test:

i) n_{01}: number of samples misclassified by the proposed system, but not by the baseline.

ii) n_{10}: number of samples misclassified by the baseline system, but not the proposed one.
iii) Null Hypothesis $\mathcal{H}_0$: The two systems perform the same.
iv) Alternative Hypothesis $\mathcal{H}_1$: The proposed system performs better.

It turned out that the test statistic Z^2 approximately follows the χ^2 distribution with 1 degree of freedom. After counting n_{01} and n_{10} from the recognition results, we looked up the test statistic in the χ^2 table. We conclude by stating that the performance gains are statistically significant at a confidence level of 99%.

7. Conclusions

Traditional linear feature transforms such as PCA and LDA, usually assume conditions that are hardly satisfied in real problems. In this work, we show how to adopt a non-linear feature transform in the HMM-based handwriting recognition systems, which has proven to be useful in automatic speech recognition. Comparing with the LDA-based baseline systems, RDT-boosted systems are able to generate 4% to 7% relative gains. As for future work, we plan to apply RDT in other modalities of the HMM-based systems for handwriting recognition, e.g., writer-dependent adaptive systems, etc.

8. Acknowledgement

The authors would like to thank Tim Ng and Krishna Subramanian for helpful discussions.

References

[1] S. Austin, R. Schwartz, and P. Placeway. The forward-backward search algorithm. In *Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing*, pages 697–700, 1991.

[2] H. Cao, R. Prasad, and P. Natarajan. A stroke regeneration method for cleaning rule-lines in handwritten document images. In *Proc. of the MOCR workshop at the 10th international Conference on Document Analysis and Recognition*, 2009.

[3] J. Chen, H. Cao, R. Prasad, A. Bhadwaj, and P. Natarajan. Gabor features for offline Arabic handwriting recognition. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 53–58, Boston, United States, June 2010.

[4] T. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.

[5] J. Droppo, L. Deng, and A. Acero. Evaluation of the splice algorithm on the Aurora2 database. In *Proceedings of the 2001 Interspeech*, pages 217–220, 2001.

[6] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[7] J. Favata and G. Srikantan. A multiple feature/resolution approach to handprinted digit and character recognition. *International Journal of Image Systems and Technology*, 7(4):304–311, 1998.

[8] M. Gales. *Model-based techniques for noise robust speech recognition*. PhD thesis, Cambridge University, 1995.

[9] X. Huang and M. Jack. Semi-continuous Hidden Markov Models for speech recognition. *Computer Speech and Language*, 3:406–408, 1989.

[10] N. Kumar and A. Andreou. A generalization of linear discriminant analysis in maximum likelihood framework. In *Proceedings of the 1996 Joint Meeting of American Statistical Association*, 1996.

[11] N. Kumar and A. Andreou. Heteroscedastic discriminant analysis and reduced rack HMMs for improved speech recognition. *Speech Communication*, 26:283–297, 1998.

[12] Z. Lu, R. Schwartz, P. Natarajan, I. Bazzi, and J. Makhoul. Advances in the BBN BYBLOG OCR system. In *Proceedings of the 1999 International Conference on Document Analysis and Recognition*, pages 337–340, 1999.

[13] P. Natarajan, Z. Lu, I. Bazzi, R. Schwartz, and J. Makhoul. Multi-lingual machine printed OCR. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):43–63, 2001.

[14] P. Natarajan, S. Saleem, R. Prasad, E. MacRostie, and K. Subramanian. Multi-lingual offline handwriting recognition using Hidden Markov Models: a script-independent approach. *Arabic and Chinese Handwriting Recognition*, pages 231–250, 2008.

[15] L. Nguyen, T. Anastasakos, F. Kubala, C. LaPre, J. Makhoul, R. Schwartz, N. Yuan, G. Zavaliagkos, and Y. Zhao. The 1994 BBN/BYBLOS speech recognition system. In *ARPA Spoken Language Systems Technology Workshop*, pages 77–81, 1995.

[16] D. Povey. *Discriminative training for large vocabulary speech recognition*. PhD thesis, Cambridge University, 2004.

[17] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig. fmpe: discriminatively trained features for speech recognition. In *Proceedings of the 2005 International Conference on Acoustics, Speech, and Singal Processing*, pages 961–964, 2005.

[18] D. Povey and P. Woodland. Minimum phone error and i-smoothing for improved discriminative training. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 105–108, 2002.

[19] R. Schwartz, L. Nguyen, and J. Makhoul. Multiple-pass search strategies. In C. Lee and F. Soong, editors, *Automatic speech and speaker recognition: advanced topics*, pages 429–456. Kluwer Academic Publishers, 1996.

[20] Y. Tay, P. Lallican, M. Khalid, C. Viard-Gaudin, and S. Knerr. An offline curisve handwritten word recognition system. In *Proceedings of the 2001 IEEE Region 10 International Conference*, pages 519–524, 2001.

[21] B. Zhang. *Discriminative feature optimization for speech recognition*. PhD thesis, Northeastern University, 2007.

[22] B. Zhang, S. Matasoukas, and R. Schwartz. Discriminatively trained region dependent feature transforms for speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 1520–1523, 2006.