Character Type Classification via Probabilistic Topic Model

Takuma Yamaguchi and Minoru Maruyama

Department of Information Engineering
Shinshu University, Nagano, 380-8553, Japan
s07t213@gmail.com, maruyama@cs.shinshu-u.ac.jp

Abstract

In this paper, we propose a method for character type classification based on a probabilistic topic model. The topic model is originally developed for topic discovery in text analysis using bag-of-words representation. Recent studies have shown the model is also useful for image analysis. We adopt the probabilistic topic model for character type classification. In our method, character type classification is carried out by classifying image patches based on their topic proportions. Since the performance of the method depends on a visual vocabulary generated by image feature extraction, we compare several feature extraction and description methods, and examine the relations to classification performance. In addition, by extending the method, we propose a coarse-to-fine approach to achieve stable character type classification for a small image patch. For that purpose, firstly, we partition an image into several patches which contain enough information to estimate the model parameters via EM algorithm. Then, each patch is subdivided into smaller patches. Estimation on the small patch is carried out by MAP-technique with a prior reflecting topic proportion of its parent patch. Through the experiments, we show accurate character type classification is made possible by the probabilistic topic model.

Keywords: *Character type classification, Probabilistic topic model, pLSA model, Bag-of-words*

1. Introduction

For document image analysis, it is often necessary to recognize “parts”' that constitute the document image. To detect such building blocks of a document image, much research has been done on layout analysis [10, 16]. Among such building blocks, some types of “parts” such as figures, tables, etc. can be extracted by recognizing geometric primitives. Besides these geometrically recognizable parts, for document analysis, it is often desirable to further categorize the text regions into more detailed classes. It often happens that several types of characters appear on a single page, such as Japanese document containing English phrases and math formulae.

In order to achieve accurate recognition, it is desirable to use type-specific method for each character type. Therefore, character type classification is useful as prior processing for document understanding. Moreover, it is possible to retrieve document images using character type information without OCR. Previous works on this subject mostly deal with binary classification problem, such as handwritten/printed and Latin/non-Latin identification [11, 14]. In this paper, we propose a method of multi character types classification via probabilistic topic model [1, 7, 8, 9].

Recently the probabilistic topic models have been applied to visual pattern recognition problems, such as image categorization, image annotation and etc. [2, 5, 17]. In our work, we use pLSA(probabilistic latent semantic analysis) model [7, 8, 9] as the probabilistic topic model. Since the probabilistic topic models are originally used in the field of document text analysis, they are defined on words, documents, and corpora. To apply the models to image analysis, we have to obtain visual words which are counterparts of ordinary "words" in text documents. Since image analysis performance depends on the feature extraction method, we compare several feature point detection and description methods, and examine the relations to classification performance.

In our work, document images are divided into small patches, and we treat them as visual documents. Character type recognition is realized by classifying the visual documents. We also propose a coarse-to-fine approach to achieve stable character type classification for a small patch. If an image is partitioned into very small patches at the beginning, each patch may include only a small number of visual words. So the classification accuracy of such a patch may be unsatisfactory using commonly-used classifier such as k-NN(k-nearest neighbour) and SVM(support vector machine) [19], because of the insufficient number of visual words in the patch. To overcome the difficulty, instead of classifying small patches directly, we adopt coarse-to-fine method. Firstly, we partition images into patches which are big enough to contain sufficient information to estimate the model parameters adequately. Then, its subdivided smaller patches are classified using the model parameters as prior information. In our experiments, we also show the pLSA-based method can give good character type classification results.

2. Probabilistic Topic Model

In this paper, pLSA (probabilistic latent semantic analysis) model [7, 8, 9] is used as the probabilistic topic model. This model is a generative statistical model for text analysis. The model is used to discover topics in a document with the bag-of-words document representation.

Let $D = \{d_1, \dots, d_M\}$ be a collection of M documents. Each document d_i includes N_i words.

A word w is an element of the vocabulary $w \in \{1, \dots, V\}$. Additionally, there is a hidden (latent) topic variable $z \in \{1, \dots, T\}$ associated with each occurrence of a word w in a document d. Where V and T represent vocabulary size and the number of topics, respectively. The pLSA model is parameterized by $\phi_{w_{in}}^{(j)}\left(= p(w_{in} \mid z = j)\right)$ and $\theta_j^{d_i}\left(= p(z = j \mid d_i)\right)$ where w_{in} is the nth word in the ith document d_i. A document is generated as follows:

1. A document d_i is selected with probability $p(d_i)$.
2. For each word in the document, a topic j is selected based on $\theta_j^{(d_i)}$.
3. A word w_{in} is generated with probability $\phi_{w_{in}}^{(j)}$.

It is assumed that the distribution of words given a latent topic z, and $\phi_{w_{in}}^{(j)}$ is conditionally independent of the document. The probabilistic graphical model of pLSA is shown in Figure 1. By marginalizing over topics z, following joint probability is obtained:

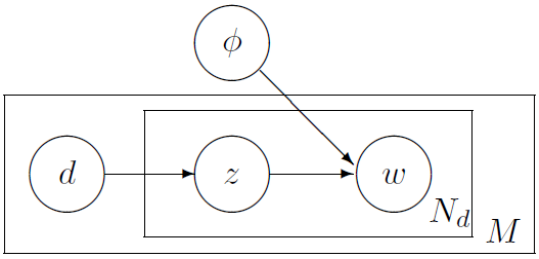


Figure 1. Graphical Model Representation of pLSA

$$p(w_{in}, d_i) = p(d_i) \sum_{j=1}^{T} \phi_{w_{in}}^{(j)} \theta_j^{(d_i)} \tag{1}$$

The model parameters $\phi_{w_{in}}^{(j)}$ and $\theta_j^{(d_i)}$ can be estimated by maximizing the data log-likelihood using an Expectation Maximization (EM) algorithm [4]. The log-likelihood is given by

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^{M} \sum_{n=1}^{N_i} \log\, p(w_{in}, d_i) = \sum_{i=1}^{M} N_i \log p(d_i) + \sum_{i=1}^{M} \sum_{n=1}^{N_i} \log \sum_{j=1}^{T} \phi_{w_{in}}^{(j)}\, \theta_j^{(d_i)} \\ &= \sum_{i=1}^{M} N_i \log p(d_i) + \sum_{i=1}^{M} \sum_{v=1}^{V} m_{iv} \log \sum_{j=1}^{T} \phi_v^{(j)} \theta_j^{(d_i)} \end{aligned} \tag{2}$$

where m_{iv} stores the number of occurrences of a word v in document d_i. EM algorithm has two steps. The first is an expectation step (E-Step), where posterior probabilities are computed for the latent variables, based on the current estimates of the parameters. The second is a maximization step (M-Step), where parameters are updated based on the so-called expected complete data log-likelihood which depends on the posterior probabilities computed in the E-Step. The EM algorithm for pLSA is:

E-Step:

$$z_{ivj} = \frac{\phi_v^{(j)} \theta_j^{(d_i)}}{\sum_{j=1}^{T} \phi_v^{(j)} \theta_j^{(d_i)}} \tag{3}$$

M-Step:

$$\phi_v^{(j)} = \frac{\sum_{i=1}^{M} m_{iv} z_{ivj}}{\sum_{v=1}^{V} \sum_{i=1}^{M} m_{iv} z_{ivj}} \tag{4}$$

$$\theta_j^{(d_i)} = \frac{\sum_{v=1}^{V} m_{iv} z_{ivj}}{\sum_{j=1}^{T} \sum_{v=1}^{V} m_{iv} z_{ivj}} \tag{5}$$

After training, the estimated parameter $\phi_v^{(j)}$ is used to calculate topic variable $\theta_j^{(d_{new})} (= p(z = j|d_{new}))$ for a novel document d_{new} through a folding-in heuristics [7, 8, 9]. In the folding-in process, EM algorithm is used in a similar manner to the training process. The folding-in steps for pLSA are:

E-Step:

$$z_{ivj} = \frac{\phi_v^{(j)} \theta_j^{(d_{new})}}{\sum_{j=1}^{T} \phi_v^{(j)} \theta_j^{(d_{new})}} \tag{6}$$

M-Step:

$$\theta_j^{(d_{new})} = \frac{\sum_{v=1}^{V} m_v z_{vj}}{\sum_{j=1}^{T} \sum_{v=1}^{V} m_v z_{vj}} \tag{7}$$

where $\phi_v^{(j)}$ is kept fixed.

In order to avoid over-fitting, we estimate the $\phi_v^{(j)}$ and θ_j^d to maximize the posterior probability, instead of maximizing the log-likelihood. We suppose Dirichlet distribution as the prior for the parameters:

$$\phi_v^{(j)} \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_V), \quad \phi_j^{(d)} \sim \mathrm{Dir}(\beta_1, \ldots, \beta_T) \tag{8}$$

In this case, the E-Step for training is the same as the equation(3), on the other hand, M-Step is:

$$\phi_v^{(j)} = \frac{(\alpha - 1) + \sum_{i=1}^{M} m_{iv} z_{ivj}}{V(\alpha - 1) + \sum_{v=1}^{V} \sum_{i=1}^{M} m_{iv} z_{ivj}} \tag{9}$$

$$\theta_j^{(d_i)} = \frac{(\beta - 1) + \sum_{v=1}^{V} m_{iv} z_{ivj}}{T(\beta - 1) + \sum_{j=1}^{T} \sum_{v=1}^{V} m_{iv} z_{ivj}} \tag{10}$$

where, for the hyper-parameters of Dirichlet distribution, we assume that $\alpha = \alpha_1 = \cdots = \alpha_V$, $\beta = \beta_1 = \cdots = \beta_T$. In the folding-in process, the E-Step is same as the equation(6), and the M-Step is given by

$$\theta_j^{(d_{new})} = \frac{(\beta - 1) + \sum_{v=1}^{V} m_v z_{vj}}{T(\beta - 1) + \sum_{j=1}^{T} \sum_{v=1}^{V} m_v z_{vj}} \tag{11}$$

3. Applying Probabilistic Topic Model to Images

3.1. Bag-of-visual Words Representation and Visual Document

Since the probabilistic topic model is originally developed in the field of document analysis, they are defined on words, documents, and corpora. Document data is represented by frequency histogram of vocabulary. To apply the model to image analysis, image data should be represented by frequency histogram of visual vocabulary in a similar way. The representation is called as bag-of-visual words or bag-of-features [3]. The image representation, which consists of a set of visual words, is derived through extracting feature points in an image, and then describing the appearance around the feature points. The visual vocabulary is obtained by vector quantization of image features. We use *k*-means algorithm for the vector quantization. First, *k*-means algorithm is applied to image features of all training data. The features in a same cluster are treated as same visual words. The center of each cluster is the representative of the visual word. Therefore, the number of clusters is the

size of the visual vocabulary. Testing data is transformed to bag-of-words representation using the visual vocabulary which obtained from training data.

In this paper, images are divided into small patches. We use both *k*-means clustering and grid-based method for grouping feature points. *K*-means clustering is applied to the set of feature points $\{x_i, y_i\}$. In the grid-based method, a document image is divided into a set of windows.

3.2. Feature Extraction

To classify character types via pLSA-based classifier, visual words are detected through image feature extraction. The image representation is derived through detecting feature point and describing the appearance around the feature points.

3.2.1. Feature Point Detection: In this research, three different types of feature extraction methods are examined. The first is a dense feature extraction by sliding a window. The window is sampled at every 4 pixels. The center of the window is selected as a feature point. If there is no black pixels in a window, the feature detection is discarded. Many scale and affine invariant feature point detectors have been recently proposed. Lindeberg [12] has developed a scale invariant interest point detector based on a maximum of the normalized Laplacian in scale space. Lowe [13] approximates the Laplacian with difference of Gaussian (DoG) filters and also detects local extrema in scale space. We use the DoG detector as the second detector.

The third feature point detector we have used is Harris-affine [15]. The detecting algorithm relies on the combination of corner points detected thorough Harris corner detection [6], multi-scale analysis through Gaussian scale-space and affine normalization using an iterative affine shape adaptation algorithm [12]. Software of Harris-affine interest point detector is available from the Oxford University Visual Geometry Group (http://www.robots.ox.ac.uk/~vgg/research/affine/). The method can give rise to the stable sparse representation which is expected to be robust to changes in scale and translation.

3.2.2. Feature Description: Along with three feature detectors mentioned above, we compared three different representations for describing appearance around the feature points. They are Haar wavelet [18], SIFT descriptor [13], and Gradient Orientation Histogram (GOH) descriptor. Haar wavelet is a multi-resolution image representation often used for object recognition. The two dimensional Haar decomposition of a square image with n^2 pixels consists of n^2 wavelet coefficients. Since we use a 16×16 pixels search window, it is represented as a 256 dimensional vector.

The SIFT descriptor is derived from windowed histograms of gradient magnitudes at varying locations and orientations, normalized to correct for contrast and saturation effects. This approach provides some invariance to lighting and pose changes. We use 128 dimensional SIFT descriptor.

The GOH descriptor is obtained in similar manner to SIFT descriptor. The difference between GOH and SIFT is rotation invariability. SIFT descriptor without orientation assignment process is GOH descriptor in our work. The dimensionality of GOH feature vector is also 128.

4. Character Type Classification

4.1. Character Type Classification Method by Probabilistic Topic Model

The pLSA is an unsupervised learning model. We may classify a novel visual document using its topic proportion $\boldsymbol{\theta}^{(d_{new})}$ estimated in the folding-in procedure. However there is no guarantee that our expected categories are the same as the topics. We assume that each category in a visual document consists of words from multiple topics.

When a novel document d_{new} is given, classification is carried out based on the probability $p(c = \ell \mid d_{new})$, which is derived via pLSA model. A visual document is assigned to the category ℓ with highest probability $p(c = \ell \mid d_{new})$, which is computed as:

$$p(c = \ell \mid d_{new}) = \sum_{j=1}^{T} \theta_j^{(d_{new})} \kappa_\ell^{(j)} \tag{12}$$

where we assume that $\kappa_\ell^{(j)} = p(c = \ell \mid z = j, d_{new}) = p(c = \ell \mid z = j)$ holds for any document *d*. For novel document d_{new}, $\theta_j^{(d_{new})}$ is obtained by fold-in procedure described in section 2. $\kappa_\ell^{(j)}$ is estimated from labeled examples as follows:

$$\kappa_\ell^{(j)} \propto p(z = j \mid c = \ell)\, p(c = \ell) \tag{13}$$

We approximate $p(z = j \mid c = \ell)$ and $p(c = \ell)$ by:

$$p(c = \ell) \approx \frac{N_\ell}{N}, \quad p(z = j \mid c = \ell) \approx \frac{1}{N_\ell} \sum_{\{i \mid category(d_i) = \ell\}} \theta_j^{(d_i)} \tag{14}$$

where N_ℓ is the number of documents of category ℓ, and *N* is the total number of examples. This classification method can be also applicable to other probabilistic topic models, such as LDA(latent Dirichlet allocation). Figure 2 shows overview of character type classification via pLSA model. Explanation for each step is described as follows:

T1. Feature extraction from training set.

T2. Generating visual vocabulary through vector quantization using extracted feature vectors.

T3. Dividing training images.

T4. Converting the divided images to visual documents based on visual vocabulary.

T5. Estimating parameters $\phi_v^{(j)}$ and $\kappa_\ell^{(j)}$.

S1. Feature extraction from a novel image.

S2. Dividing the input image.

S3. Creating visual documents from the divided images based on visual vocabulary.

S4. Character type classification for each visual document.

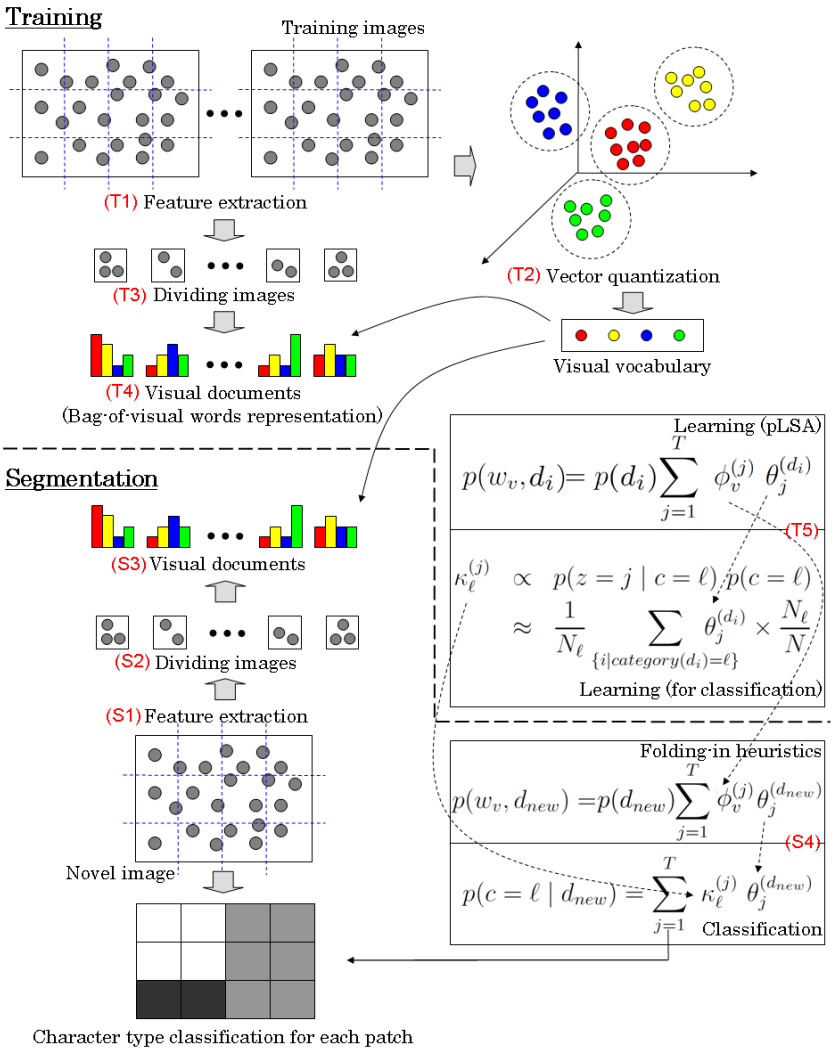


Figure 2. Overview of Character Type Classification using pLSA Model

4.2. Classification of Small Visual Document through Coarse-to-fine Approach

In the previous subsection, we have proposed a method to estimate category of a given visual document. For the method to work properly, it is necessary there is enough number of visual words within the document. When the number of visual words in the document is too few, we cannot expect to obtain accurate category label. The area corresponding to the visual document should not be too small to obtain reliable categorization result. This implies, it is difficult to obtain fine and reliable character type classification by the method described so far.

To overcome the difficulty, we propose a coarse-to-fine approach for fine and accurate character type classification. If we divide a document image into very small patches and then try to estimate the category of each region, it is difficult to obtain reliable results due to lack of information (i.e. sufficient number of visual words) within the region. In our method, instead of dividing a document image into very small patches, it is divided so that the area of each patch is large enough to contain sufficient number of visual words. For each patch d_{new}, parameter $\boldsymbol{\theta}^{(d_{new})}$ is calculated by the method described in the previous subsection. Then,

each patch is subdivided into small children d'_{new} and categorization is carried out based on the estimation result on the parent patch.

As described before, $\boldsymbol{\theta}^{(d_{new})}$ is estimated based on maximization of posterior probability. For that purpose, simple Dirichlet prior $\mathrm{Dir}(\beta_1, \ldots, \beta_T)$, $\beta_1 = \cdots = \beta_T = \beta$ is used. This prior reflects our prior knowledge about the topic distribution in the parent patch. Fold-in procedure for estimation of topic distribution in the small child patch $\boldsymbol{\theta}^{(d'_{new})}$ also proceeds similarly. In the procedure, however, instead of using very simple hyper parameter $\beta = \beta_1 = \cdots = \beta_T$, we use the estimated parameter $\boldsymbol{\theta}^{(d_{new})}$ of the parent patch. The Dirichlet prior and the EM algorithm are given as :

$$\theta_j^{(d'_{new})} \sim \mathrm{Dir}(\beta'_1, \ldots, \beta'_T), \quad \beta'_j = \beta_0 + \gamma\theta_j^{(d_{new})} \tag{15}$$

E-Step:

$$\hat{z}_{d'_{new}vj} \propto \phi_v^{(j)}\theta_j^{(d'_{new})} \tag{16}$$

M-Step:

$$\theta_j^{(d'_{new})} \propto \sum_{v=1}^{V} m_{d'_{new}v}\,\hat{z}_{d'_{new}vj} + \beta_0 + \gamma\theta_j^{(d_{new})} - 1 \tag{17}$$

where, γ is the coefficient to control the contribution of the estimation on the topic distribution of the parent patch. The category ℓ of $\mathrm{d}'_{\mathrm{new}}$ can be decided by:

$$p(c = \ell \mid d'_{new}) = \sum_{j=1}^{T} \theta_j^{(d'_{new})}\kappa_\ell^{(j)} \tag{18}$$

Even if the number of visual words is not many, a stable classification result is obtained with our approach, because the parent topic distribution which is estimated with enough visual words is reflected in the prior.

5. Experiments

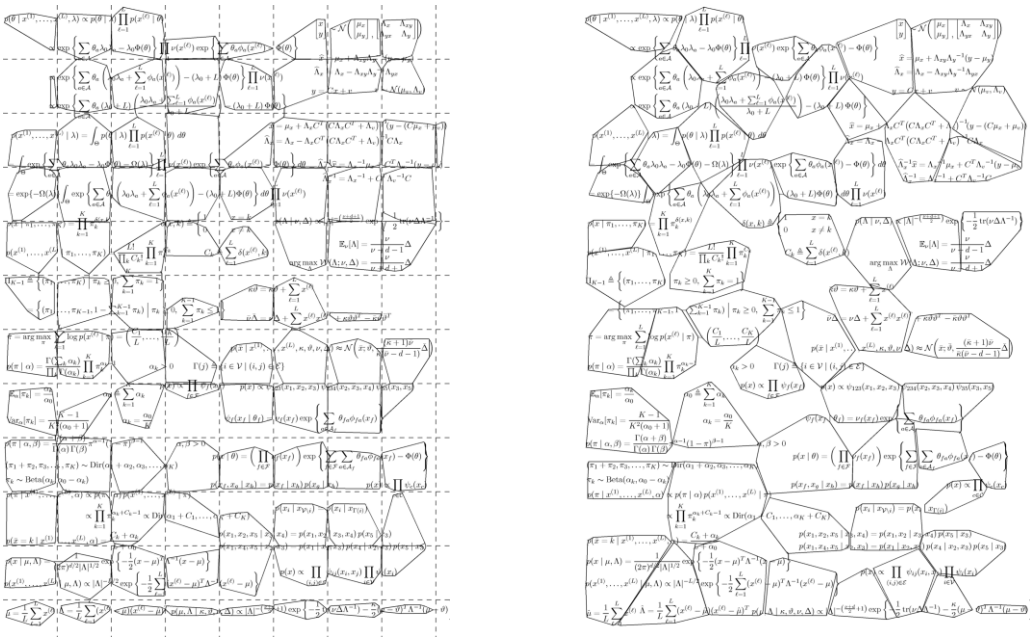
5.1. Data Sets

In this paper, we treat document images which consist of text regions of the following four categories: mathematical formula, printed Japanese, printed English and handwritten. We collected images of scientific papers written in printed Japanese, printed English and handwritten. These images were scanned with 300 dpi. For each category, we synthesized sample images by editing these raw images so that each resultant image just contains elements only from the corresponding category. For example, math formula images were made by collecting many mathematical formula regions from many papers. Likewise, for example, printed Japanese document images were made so that they contain only printed Japanese characters and numeric symbols included in printed Japanese strings. In Table 1, we show categories and corresponding characters included in the text regions. For each edited document image, feature points and the descriptions are detected using DoG detector and SIFT descriptor, respectively. Then, visual documents are derived by dividing the images. To divide the image, we examined the following two methods. The examples of the two types of

visual documents are shown in Figure 3. In Table 2, the numbers of extracted regions of the four categories are shown.

Table 1. Character Types Contained in each Category

Categories	Contained character types
Mathematical formula	Mathematical expression English / numeric in mathematical formula
Printed Japanese	Printed Japanese Numeric characters in Japanese strings
Printed English	Printed English English / numeric characters in English strings
Handwritten	Handwritten characters



Regular grid-based regions. *K*-means-based regions.

Figure 3. Two Types of Deviation

Table 2. The Number of Images and Local Regions of each Category

	# of images	*k*-means method	300×300 pixels grid	240×240 pixels grid
Math formula	12	853	1,014	1,514
Printed Japanese	34	2,599	1,408	2,170
Printed English	15	1,797	797	1,193
Handwritten	42	2,779	2,332	3,394
	103	8,028	5,551	8,271

1. Regular grid

   Each image is divided into cells by using regular grids. 300×300 pixels cells and 240×240 pixels cells are used. In the experiments, we reject cells which contain less than 25 visual words.

2. *K*-means method

   Since the grid-based method does not care about the distribution of feature points in a document image, the number of feature points (i.e. visual words) included in each cell could vary. To reduce the variation, we divide text image into sub-regions by *k*-means method so that each region contains approximately same number of feature points. *K*-means method is applied to the extracted interest points based on their distances. The resultant clusters are treated as visual documents. The number of clusters *k* is manually determined so that the numbers of feature points within the clusters are similar.

All experimental results in this paper are the average of 5 times experiments with different data sets.

5.2. Classification with pLSA Model

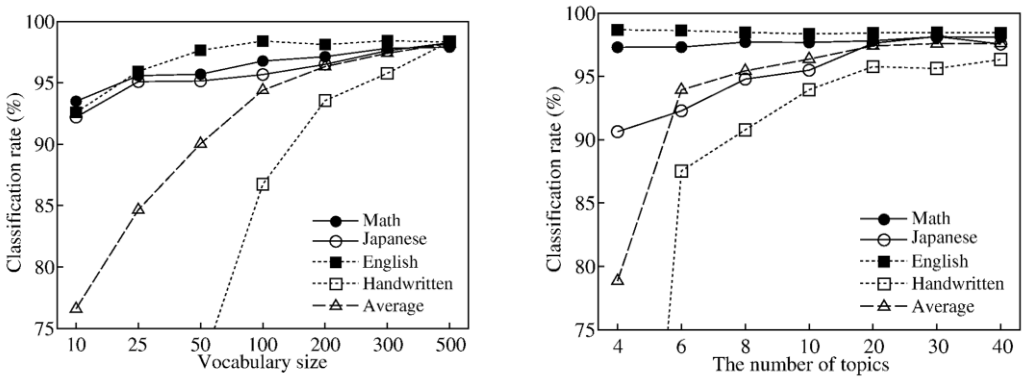
5.2.1 Determination of pLSA Model Parameters: To use the pLSA model, we need to decide the vocabulary size of bag-of-visual words representation, and the number of topics. In order to decide these parameters, we investigated classification performance with respect to the vocabulary size and the number of topics.

For learning data, we sampled 100 patches for each category randomly from the data set generated by *k*-means method. The combination of DoG and SIFT was used as the feature extraction method. Visual vocabulary was created from the feature vectors in the patches using vector quantization. For test data, 8,271 patches obtained by grid method were used. The pixel size of each patch in test data set was 240×240.

MAP-based EM algorithm described in section 2 was applied to these data sets. Hyper parameters (parameters of Dirichlet priors) we used were $\alpha = 2$, $\beta = 2$ (see eqs.(9), (10)). Figure 4 shows the classification results. The results show the larger the vocabulary size, the higher the classification rate, and the more the number of topics, the better the rate. It is also shown, apparently good results were obtained when the number of topics was more than the number of categories.

Among all the categories, classification results for handwritten was not good, especially with small vocabulary size and small number of topics. This is because the handwritten patches mainly consist of handwritten Japanese. It is hard to discriminate between printed Japanese and handwritten Japanese. To classify them correctly, it is preferable to have large vocabulary and a large number of topics.

Based on these experimental results, we let the vocabulary size and the number of topics be 300 and 20, respectively throughout all the experiments to be described below.



(a) (b)

Figure 4. Comparison of the Classification Rates with Respect to (a) Size of Visual Vocabulary and (b) the Number of Topics

5.2.2 Comparison of Methods for Generating Visual Documents: We compared two types of methods for dividing a document image into image patches described in 5.1. They are grid-based method and *k*-means based method. Apparently, grid-based method is much easier to implement. However, with the grid method, the number of feature points included in the patches may vary significantly. It depends on the input document images. The imbalance may cause the problems for categorization. On the other hand, with the *k*-means method, the number of the features in each patch is kept approximately constant. To evaluate effect of the patch generation method, we compared the classification rate for the data set by grid-based method and the rate for the data set by *k*-means method. Figure 5 shows the results. As the figure shows, although the difference between the rates is not so large, *k*-means method outperformed the grid method

5.2.3 Comparison of Classification Methods: In order to examine the classification capability of the probabilistic topic model (pLSA) for image patches, we compared its performance with three other classification methods, *k*-nearest neighbor (*k*-NN) algorithm with Euclidean distance function, SVM (support vector machine), and LDA (latent Dirichlet allocation)-based method. For *k*-NN and SVM, we have used normalize histogram of occurrence. Let n_v be the number of occurrence of visual word v in an image patch. Normalized histogram of visual words is defined as $\boldsymbol{h} = (h_{1,} \dots, h_V)^T, h_v = n_v / \sum_{v=1}^{V} n_v$ where V is a vocabulary size. SVM and *k*-NN are applied using the normalized histogram.

With *k*-NN method, each novel visual document is classified by a majority vote of its neighbors. It is assigned to the category most common among its *k*-nearest training samples. SVM classifies the input vector $\boldsymbol{h}$ by the following function.

$$f(\boldsymbol{h}) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{h}, \boldsymbol{h}_i) - b$$

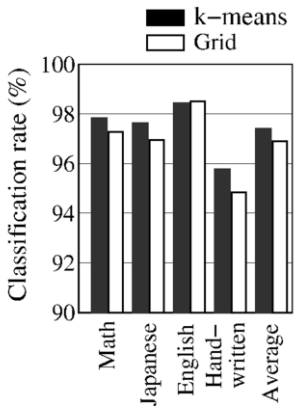


Figure 5. Comparison of the Classification Rates based on *k*-means- and grid-based Patch Generation

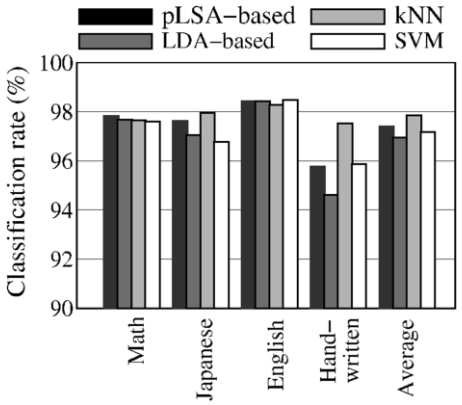


Figure 6. Classification Rates of each Classifier

where $K(\boldsymbol{u}, \boldsymbol{v})$ is a kernel function. Coefficients $\{\alpha_i\}$ are non-zero only for the subset of the training samples called support vectors. The performance of SVM depends on the choice of kernel function. In our work, we have used Gaussian kernel, which outperformed the other commonly used kernels in the preliminary experiments. To apply SVM to multi-category classification problems, we employ one-vs-rest method. For each category, an SVM is trained to discriminate the category from the others. Then, for novel input, the best category is decided by the SVM that gives the highest output value.

LDA is a probabilistic topic model like pLSA. Unlike pLSA, in LDA, topic proportions are considered as random variables. Let $\theta = (\theta_1, \ldots, \theta_T)$ be topic proportions. Generative process for each visual document is as follows :

1. For each topic $j \in \{1, \ldots, T\}$, word probabilities $\widetilde{\boldsymbol{\phi}}^{(j)} = (\tilde{\phi}_1^{(j)}, \ldots, \tilde{\phi}_V^{(j)})$ are chosen based on Dirichlet distribution:

$$\widetilde{\boldsymbol{\phi}}^{(j)} \sim \mathrm{Dir}\left(\widetilde{\boldsymbol{\phi}} \mid \alpha_1, \ldots, \alpha_V\right)$$

2. Topic proportions $\widetilde{\boldsymbol{\theta}}^{(d)} = (\tilde{\theta}_1^{(d)}, \ldots, \tilde{\theta}_T^{(d)})$ are chosen for each document according to Dirichlet distribution:

$$\widetilde{\boldsymbol{\theta}}^{(d)} \sim \mathrm{Dir}\left(\widetilde{\boldsymbol{\theta}} \mid \beta_1, \ldots, \beta_V\right)$$

3. For each word in the document *d*

(a) Choose topic z_n according to the multinomial distribution:

$$z_n \sim \mathrm{Mult}(z \mid \widetilde{\boldsymbol{\theta}}^{(d)})$$

(b) Choose a word w_n according to the multinomial distribution :

$$w_n = \mathrm{Mult}(w \mid \widetilde{\boldsymbol{\phi}}^{z_n})$$

In pLSA, although the number of unknown parameters grows with sample size, in LDA, the number of parameters does not grow because topic proportions are treated as random variables. However, although the inference algorithm (EM algorithm) for pLSA is easy to implement, for posterior inference under LDA model, more complicated approximate inference algorithm such as MCMC (Markov chain Monte-Carlo) or variational methods must be exploited.

The results are shown in Figure 6. The results indicate the pLSA-based method can perform similarly well compared to the other classification methods. The experiment shows pLSA, which is very much easier to implement, has enough categorization power compared to LDA. Compared to the other types of classification methods, it has been shown that pLSA has enough classification ability. We will show later (see 5.4), proposed method (pLSA-based method) can outperform the others for small image patches.

5.3 Selection of Feature Detectors and Descriptors and its Effect Against Geometric Transformations

To apply the probabilistic topic model to image patches, we have to define visual words using the feature detector and descriptor. In the previous experiments, we have used DoG detector + SIFT descriptor. In this subsection, we examine the effect of detector and descriptor selection against the scale change and rotation of image patches. For the experiments, scaling and rotation operations have been applied to image sets. Scaling factors we have used are: 0.75, 1.0(original, unchanged), and 1.25. Rotation angles (degree) we have used are: 0 (original, unchanged), 5, 10, and 20. These transformations were applied to the test data set. Training data set is generated without these geometric transformations. The number of image patches for learning is 400 (100 for each category). The number of patches for evaluation is 2,000 (500 for each category).

For feature detection, we use DoG and Harris-affine, which are known to be robust to changes in scale (DoG is scale invariant, and Harris-affine is affine invariant). They produce sparse feature points. In addition to DoG and Harris-affine detectors, we have examined a feature detection method by sliding window, which gives dense feature points. In the experiment, the window was sampled at every 8 pixels. We have compared these three descriptors with respect to scale changes in images. For feature description, we used SIFT. In Figure 7 and Table 3, the classification results are given. The Table 3 shows the changes in classification rates (ratio of classification rates). As expected, the scale invariant detector DoG and affine invariant detector Harris-affine showed better performance against the scale change. The results indicate, it is better to use invariant detector such as DoG and Harris-affine, especially in the case where font size could be different between training samples and the target documents.

Next, we have examined classification results for rotated image patches. We have used SIFT (rotation invariant), GOH and Haar wavelet as feature descriptors along with DoG and sliding-window based dense feature detectors. In Figure 8, classification results are shown. Among the 5 combinations of detector and descriptor, both DoG + SIFT and dense detector + SIFT outperformed the others. These experimental results indicate combination of invariant feature detector and descriptor is preferable for visual document generation as expected.

Table 3. Comparison of Performance Changes of Classification Rates by scale Changes

	×0.75	×1.0	×1.25
DoG + SIFT	0.957	1.0	0.971
Harris affine + SIFT	0.99	1.0	0.955
Dense + SIFT	0.939	1.0	0.933

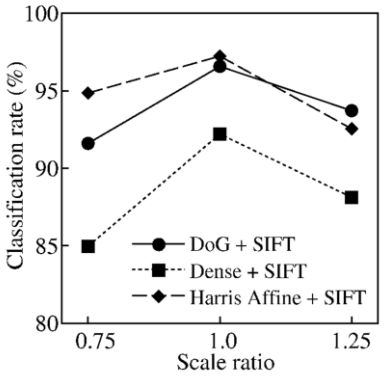


Figure 7. Classification Results for Scale Changed Data

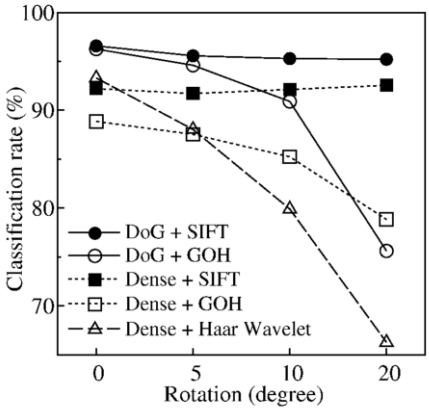


Figure 8. Classification Results for Rotated Data

5.4 Coarse-to-fine Approach for a Small Image Patch

In the previous experiments, we used image patches which contain sufficient number of visual words. However, in practice, if the image patch is large, it may contain regions of different kinds of categories. It is difficult to categorize such an image patch. To obtain fine character type information, smaller image patches are preferable. However, reducing image patch size leads to lack of visual words within a patch, and thus it brings unstable categorization. To overcome the difficulty, in 4.2 we have proposed a coarse-to-fine classification method. We examine the effectiveness of the proposed method by experiments.

To evaluate the proposed method, we synthesized image patches each of which consisted of sub-patches of multiple categories. In practice, it seems to be rare that a patch consists of the combination of more than three categories. We synthesized a set of parent patches described in 4.2 by sampling small patches in two different categories. We have examined 5 types of mixing ratio; 9:1, 8:2, 7:3, 6:4 and 5:5. For each type, we generated 3,000 parent visual documents, each of which contained 20 children. In addition, we prepared 1,000 parent visual documents, each of which contained 20 children from a single category (i.e. 10:0). For feature detection and description, we used DoG + SIFT.

To apply the methods in 4.2, we have to specify hyper-parameter γ, which controls uncertainty of prior distribution given by a parent patch. For that purpose we examined the performance of the proposed classification method with various γ values. Each classifier was applied to the 5 types of data sets mentioned above. The classification results of small patches (children) are shown in Figure 9. Where, $\gamma = 0$ indicates the classification results

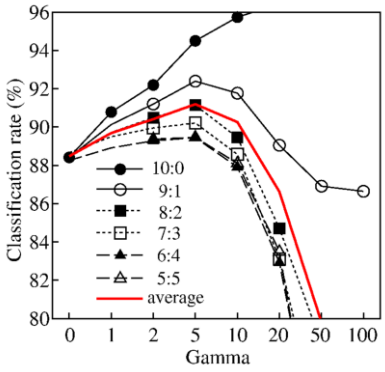


Figure 9. Experimental Results for Specification of Prior Parameter

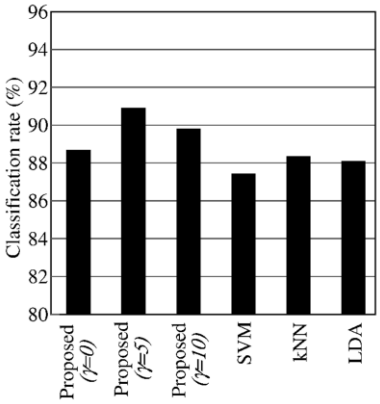
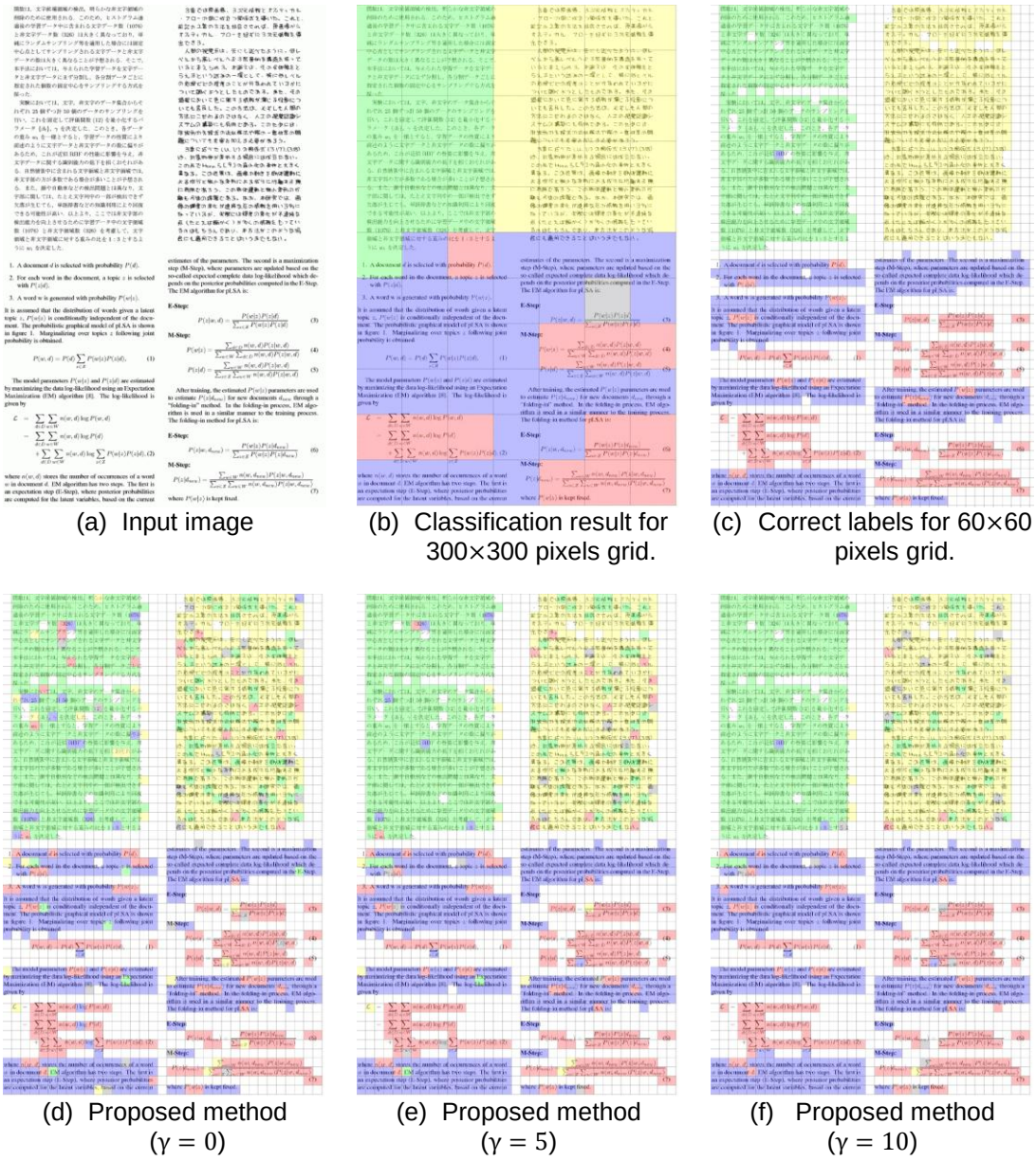


Figure 10. Comparison of Classifiers on Small Document Patch Classification

with no-prior, that is simple MAP-EM based method. We obtained the highest performance on average with $\gamma = 5$. For comparison, we also applied SVM, *k*-NN and LDA to the small visual documents classification. Figure 10 shows the average classification rates of these classifiers. The results indicate the effectiveness of the proposed coarse-to-fine method for small image patch classification.

Figures 11(a)-(f) show character type classification results for each patch in a document by using the pLSA-based classifiers described in this paper. The input image shown in Figure 11(a) was synthesized from four categories of character types in document images. Figure 11(b) shows the classification result for 300×300 pixels regular grid patches. Each patch was classified by pLSA-based classifier described in 4.1. The colors; red, green, blue and yellow indicate the parts classified as mathematical formula, printed Japanese, printed English, and handwritten, respectively. In the experiment, patches with less than 25 visual words were discarded, and such regions were shown in white (background) color. And also, if the difference between the highest and the second highest values of $p(c \mid d)$ was less than 0.01, we judged the classification result unreliable. Grey color indicates the regions where unreliable results were obtained. For comparison, in Fig.11(c), correct labels (categories) are shown. In the figure, category of each small patch, whose size is 60×60 pixels, is manually specified. We neglected patches which consist of elements of multiple categories. These figures show that although the classification result with 300×300 pixels patches is reasonably well, patch size is too large to obtain fine character type information of the document image.

Figures 11(d),(e),(f) show results of the coarse-to-fine approach described in 4.2. We generated parent patches with 300×300 pixels regular grid and then divided each parent patch into 5×5 children with 60×60 pixels patch. The result with $\gamma = 0$ in Figure 11(d) equals to the result of simple pLSA-based classifier described in 4.1, because $\gamma = 0$ negates the effect of parent topic proportion. In Figures 11(e),(f), the results by coarse-to-fine approach with $\gamma = 5$ and $\gamma = 10$ are given. These results are close to the correct one shown in Figure 11(c), and much better than the result in Figure 11(d).



(a) Input image (b) Classification result for 300×300 pixels grid. (c) Correct labels for 60×60 pixels grid.

(d) Proposed method ($\gamma = 0$) (e) Proposed method ($\gamma = 5$) (f) Proposed method ($\gamma = 10$)

Figure 11. Character Type Classification Results of a Document Image

6. Conclusion

In this paper, we have proposed a method for character type classification on a document image based on a probabilistic topic model, pLSA. In our method, character type classification is carried out by classifying image patches using their topic proportions. Since the performance of the method depends on a visual vocabulary generated by image feature extraction, we have compared several feature extraction and description methods, and examined the relations to the classification performance. We also have examined the robustness to the scale change and rotation. The experimental results indicate the visual

vocabulary that is derived by invariant detectors and descriptors is preferable for document image analysis.

By extending the pLSA-based classification method, we have proposed a coarse-to-fine approach to obtain fine character type information of document image. For that purpose, firstly, we partition an image into patches which contain enough information to estimate the model parameters via EM algorithm. Then, each patch is subdivided into small patches. Estimation on the small patch is carried out by MAP-technique with a prior reflecting its parent topic proportions. Experimental results show the proposed method can outperform other classifiers such as simple pLSA-method, LDA, *k*-NN, and SVM.

References

[1] D. Blei, A. Ng and M. Jordan, "Latent dirichlet allocation", Journal of Machine Learning Research, vol. 3, **(2003)**.
[2] A. Bosch, A. Zisserman and X. Munoz, "Scene classification via plsa", In Proceedings of the European Conference on Computer Vision, **(2006)**.
[3] G. Csurka, C. Bray, C. Dance and L. Fan, "Visual categorization with bags of keypoints", In Proceedings of ECCV Workshop on Statistical Learning in Computer Vision, **(2004)**.
[4] A. Dempster, N. Laird and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm", Journal of the Royal Statistical Society B, vol. 39, no. 1, **(1977)**.
[5] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories", In Proceedings of Computer Vision and Pattern Recognition, **(2005)**.
[6] C. Harris and M. Stephens, "A combined corner and edge detector", In In Alvey Vision Conference, **(1988)**.
[7] T. Hofmann, "Probabilistic latent semantic analysis", In Proceedings of Uncertainty in Artificial Intelligence, **(1999)**.
[8] T. Hofmann, "Probabilistic latent semantic indexing", In Proceedings of Special Interest Group on Information Retrieval, **(1999)**.
[9] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis", Machine Learning, vol. 42, **(2001)**.
[10] A. K. Jain and B. Yu, "Document representation and its application to page decomposition", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 3, **(1998)**.
[11] E. Kavallieratou, S. Stamatatos and H. Antonopoulou, "Machine-printed from handwritten text discrimination", In International Workshop on Frontiers in Handwriting Recognition, **(2004)**.
[12] T. Lindeberg and J. Garding, "Shape-adapted smoothing in estimation of 3-d shape cues from affine deformations of local 2-d brightness structure", Image and Vision Computing, vol. 15, no. 6, **(1997)**.
[13] D. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, vol. 60, no. 2, **(2004)**.
[14] H. Ma and D. Doermann, "Gabor filter based multi-class classifier for scanned document images", In International Conference on Document Analysis and Recognition, **(2003)**.
[15] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points", In Proceedings of International Conference on Computer Vision, **(2001)**.
[16] L. O'Gorman, "The document spectrum for page layout analysis", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, no. 11, **(1993)**.
[17] J. Sivic, B. Russell, A. Efros, A. Zisserman and W. Freeman, "Discovering objects and their location in images", In Proceedings of International Conference on Computer Vision, **(2005)**.
[18] E. Stollnitz, T. DeRose and D. Salesin, "Wavelet for computer graphics: A primer part 1", IEEE Computer Graphics and Applications, vol. 15, no. 3, **(1995)**.
[19] V. Vapnik, "The nature of statistical learning theory", Springer, **(1995)**.

Authors

Takuma Yamaguchi received his B.E, M.E and Ph.D. in information engineering from Shinshu University, Japan in 2002, 2004 and 2010, respectively. He had been a research scientist and a software engineer at Media Drive Corporation and Nippon Signal Corporation since 2004 until 2012. Currently, he is a software engineer in Gree, Inc. His research interests include image processing, machine learning and operations research.

Minoru Maruyama received his B.E. and Ph.D. both in mathematical engineering from the University of Tokyo in 1982 and 1993, respectively. He was with the central research laboratory, Mitsubishi Electric Corporation from 1982 to 1996. He had been an associate professor since 1996 until 2010 in the Department of Information Engineering, Shinshu University, Japan and since then, he has been a professor in the same department. He was a visiting scientist at the Artificial Intelligence Laboratory at MIT from 1990 to 1991. His research interests include computer vision, learning, and computer graphics. M. Maruyama is a member of ACM and IEEE.