Structured Feature Similarity with Explicit Feature Map

Takumi Kobayashi
National Institute of Advanced Industrial Science and Technology
Umezono 1-1-1, Tsukuba, Japan
takumi.kobayashi@aist.go.jp

Abstract

Feature matching is a fundamental process in a variety of computer vision tasks. Beyond the standard L_2 metric, various methods to measure similarity between features have been proposed mainly on the assumption that the features are defined in a histogram form. On the other hand, in a field of image quality assessment, SSIM [27] produces effective similarity between images, taking the place of L_2 metric. In this paper, we propose a feature similarity measurement method based on the SSIM. Unlike the previous methods, the proposed method is built on not a histogram form but a tensor structure of a feature array extracted such as on spatial grids, in order to construct effective SSIM-based similarity measure of high robustness which is a key requirement in feature matching. In addition, we provide the explicit feature map such that the proposed similarity metric is embedded as a dot product. It contributes to significant speedup in similarity measurement as well as to feature transformation toward an effective vector form to which linear classifiers are directly applicable. In the experiments on various tasks, the proposed method exhibits favorable performance in both feature matching and classification.

1. Introduction

It is a fundamental and primary process in computer vision tasks to match/compare features extracted from images and videos. Its application covers keypoint matching [8], retrieval [7] and classification based on exemplars such as by k-NN, while in recent years, feature matching is also found in scene parsing or flow estimation by SIFT flow [14]. The features are matched based on similarities between them, and along with the development of feature extraction methods, the similarity measurement methods are attracting keen attention.

The most standard (dis)similarity measure is L_2 metric. It is regarded as a natural choice on the basis that feature vectors are embedded in the Euclidean space. Depending on feature extraction methods, however, the feature vectors are not distributed throughout the Euclidean space but restricted in a subspace with a constraint inherently imposed by the extraction method. For example, in the computer vision community, image features are enthusiastically developed in a form of histogram comprising non-negative values. By exploiting the intrinsic characteristics of features, the (dis)similarity measurement methods are proposed beyond the L_2 metric. χ^2 distance [2] is a commonly used distance measure for histogram-based features derived from statistical χ^2 test, being also applied to a kernel function [31]. In the other approach, the Earth Mover's distance (EMD) [21] is proposed by applying an optimization problem of transportation to effectively measure dissimilarity between histogram features. The EMD, however, requires huge computational cost, which motivates to propose faster variants of EMD such as in [17, 16], and it is specialized to SIFT features as SiftDist [16]. The EMD takes into account the relationships between the histogram bins while L_2 and χ^2 metrics are composed solely of differences in corresponding bins. Such cross-bin distance is also employed in diffusion distance [13] based on the structure of histogram features other than a simple vector. Such feature structure is introduced into the proposed method as described in the later.

In the other literature than the feature matching, an image quality assessment requires such similarity metric between *images* that are close to human perception. In that field, it is widely known that L_2 metric is not compatible with the human perception and thus unsuitable for image similarity measure. According to the human visual system, structural similarity (SSIM) index [26, 27] was proposed. The method measures a similarity between a reference image and its distorted one by exploiting the structural characteristics in the image (patch) as in the human visual process. The structural information extracted by the SSIM is similar to the cross-bin distance mentioned above; we show the detailed form in Sec. 2. The SSIM has taken the place of L_2 metric in image quality assessment since it thoroughly defeats L_2 in a variety of experiments, and some variants of SSIM have also been proposed [28, 29, 3].

In this paper, based on the SSIM measure, we propose

a novel similarity metric of the features that have structure beyond one-way array (vector), not limited to a non-negative histogram form unlike the previous methods. Recent features are frequently defined in a structured array form [11, 13], though most methods unfold them into a vector; for example, local primitive features are extracted on the two-dimensional spatial positions (grids) to form a three-way *tensor* as particularly found in local descriptors such as SIFT [15] and SURF [1]. We effectively incorporate such feature structure into similarity measurement for enhancing robustness, which is demanded in feature matching, with retaining discriminative power of SSIM. In addition, we provide the explicit feature map in which the proposed similarity is embedded as a dot product. An ordinary similarity measurement operates respective pairs of features, requiring significant computation time. The explicit mapping enables us to efficiently compute the proposed similarity measure by dot products which result in matrix multiplication performed in a computationally efficient way such as by the BLAS library. Furthermore, the explicit feature map is regarded as feature transformation into an effective vector form to which linear classifiers are directly applied. Thus, the proposed method works for measuring feature similarity as well as transforming features.

2. SSIM for image quality assessment

We review the formulation of SSIM [27, 26] in image quality assessment and mention its applicability to (generic) feature matching. Given a reference image $\mathcal{I}_x$, the target (distorted) image $\mathcal{I}_y$ is assessed in terms of quality by measuring its fidelity to $\mathcal{I}_x$. The SSIM operates on an image patch pair of $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^D$ drawn from $\mathcal{I}_x$ and $\mathcal{I}_y$ to assign the following similarity measure $\mathcal{S}$:

$$\mathcal{S}(\boldsymbol{x},\boldsymbol{y}) = \mathcal{M}(\boldsymbol{x},\boldsymbol{y})\,\mathcal{V}(\boldsymbol{x},\boldsymbol{y})\,\mathcal{C}(\boldsymbol{x},\boldsymbol{y}), \tag{1}$$

$$\mathcal{M}(\boldsymbol{x},\boldsymbol{y}) = \mathtt{k}(\mathtt{u}(\boldsymbol{x}),\mathtt{u}(\boldsymbol{y})),\ \mathcal{V}(\boldsymbol{x},\boldsymbol{y}) = \mathtt{k}(\mathtt{q}(\boldsymbol{x}),\mathtt{q}(\boldsymbol{y})), \tag{2}$$

$$\mathcal{C}(\boldsymbol{x},\boldsymbol{y}) = \frac{(\boldsymbol{x}-\mathtt{u}(\boldsymbol{x}))^\top(\boldsymbol{y}-\mathtt{u}(\boldsymbol{y}))}{\|\boldsymbol{x}-\mathtt{u}(\boldsymbol{x})\|_2\|\boldsymbol{y}-\mathtt{u}(\boldsymbol{y})\|_2}, \mathtt{k}(a,b) = \frac{2ab}{a^2+b^2}, \tag{3}$$

where $\mathtt{u}(\boldsymbol{x})$ and $\mathtt{q}(\boldsymbol{x})$ are functions to compute mean and standard deviation of pixel values in patches $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and $\mathtt{k}(a,b)$ is a function to measure similarity between two scalars a and b. A similarity between two images $\mathcal{I}_x$ and $\mathcal{I}_y$ is then computed by averaging the above patch-based SSIM scores $\mathcal{S}$ over a whole image.

Three functions $\mathcal{M}, \mathcal{V}$ and $\mathcal{C}$ in (2, 3) measure similarities regarding *luminances*, *contrasts* and *structures* in the patches, respectively. The structural similarity $\mathcal{C}(\boldsymbol{x},\boldsymbol{y})$ extracts pixel relationship, correlation coefficient, as in cross-bin distance. It, however, is too robust in pixel value changes to give favorable similarity measure since it always produces maximum similarity score (*i.e.*, 1) for affine relationship between pixel values, $y_i = \alpha x_i + \beta$, $(\alpha > 0)$. To compensate it, the other two types of similarities $\mathcal{M}$ and $\mathcal{V}$ are complementarily introduced to capture changes of luminance (bias β) and contrast (scaling α). These measurements are related to a human perceptual system [26].

On the other hand, the dot product, a simple similarity measure in the Euclidean space, is decomposed into

$$\boldsymbol{x}^\top\boldsymbol{y} = D\left\{\mathtt{q}(\boldsymbol{x})\mathtt{q}(\boldsymbol{y})\mathcal{C}(\boldsymbol{x},\boldsymbol{y}) + \mathtt{u}(\boldsymbol{x})\mathtt{u}(\boldsymbol{y})\right\}, \tag{4}$$

which is different from, but related to (1). Namely, the luminance and contrast similarities degenerate into the simple products, $\mathtt{u}(\boldsymbol{x})\mathtt{u}(\boldsymbol{y})$ and $\mathtt{q}(\boldsymbol{x})\mathtt{q}(\boldsymbol{y})$, respectively. Though the luminance one is separated into an additive form, such formulation is also found in SSIM variant [3]. Thus, based on the comparison of (1) and (4), it turns out that the success of SSIM is largely due to the function $\mathtt{k}(a,b) = \frac{2ab}{a^2+b^2}$.

$\mathtt{k}(a,b)$ can be rewritten by using $\theta = \arctan(\frac{b}{a})$ as $\mathtt{k}(a,b) = \frac{2ab}{a^2+b^2} = \cos\{2(\theta - \frac{\pi}{4})\}$ which measures a difference between a and b based on the ratio $\frac{b}{a}$. Thereby, the difference $|a-b|$ contributes to the similarity $\mathtt{k}(a,b)$ differently according to $r = \sqrt{a^2+b^2}$; $\mathtt{k}(a,b)$ is vulnerable to $|a-b|$ on smaller r while it is insensitive on larger r. Even though such functionality is inspired from the human perceptual process [26], it is also compatible with generic feature similarity. It is recently shown that feature transform by squared root [22] and log [10] successfully improves performance via the similar functionality as above, increasing resolution in smaller feature values while suppressing it in larger ones; in particular, the log transform is closely related to the ratio $\frac{b}{a}$. Thus, the function $\mathtt{k}$ is considered to be useful for establishing effective feature similarity measure.

3. Structured feature similarity

Based on the above analysis of the SSIM formulation, we propose a method to measure similarity for matching features, such as SIFT [15], by leveraging the SSIM measure. The straightforward way to incorporate SSIM is to directly feed feature vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ into (1). Such naive method, however, does not work well, being inferior even to L_2 metric as will be shown in Fig. 3. This is because the feature matching is different from image quality assessment in terms of *robustness*, though both of them are built on similarity measurement. The SSIM has been successfully applied to measure degree of distortion in the target image by effectively characterizing subtle image changes. In contrast, the feature matching requires to discriminate the target itself while being highly robust to those distortions. Thus, we propose a similarity measure of features so as to enhance robustness which SSIM lacks, with retaining the discriminative power of the SSIM.

While the previous methods [2, 17, 13, 16] assume a histogram form in the features, our assumption is that the features are formulated in a *structured* form, for example,

three-way tensor. As mentioned in [11], most image features extracted on spatial domain are essentially formulated in a tensor (or matrix) rather than in a simple vector. The proposed method exploits the intrinsic feature structure and reconsiders similarity measurement functions for enhancing robustness.

3.1. Feature structure

For enhancing robustness to feature perturbations, the whole feature $\boldsymbol{x}$ is represented by an ensemble of n *sub-features* $\hat{\boldsymbol{x}}_l$ on which the similarity measure is computed and then summed up as follows:

$$\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{l=1}^{n} w_l \mathcal{S}(\hat{\boldsymbol{x}}_l, \hat{\boldsymbol{y}}_l), \tag{5}$$

where $\hat{\boldsymbol{x}}_l$ and $\hat{\boldsymbol{y}}_l$ are the l-th sub-features assigned with the weight w_l ($\sum_{l=1}^{n} w_l = 1$), and $\mathcal{S}$ is a similarity function defined in Sec. 3.2. Most of features extracted from the spatial domain (image) are intrinsically formulated in a three-way *tensor* of $I \times J \times K$, $\boldsymbol{x} = \{x_{ijk}\}_{i=1,j=1,k=1}^{I,J,K}$, where I indicates the dimensionality of local primitive feature and J, K are the number of spatial bins along x, y-axes; for example, SIFT [15] consists of $8_{(I)}$-dimensional gradient orientation histogram extracted on $4_{(J)} \times 4_{(K)}$ spatial grids. Based on the tensor structure, there are four conceivable ways to define the form of sub-features as follows (Fig. 1):

1. VECTOR: This is the same as the above-mentioned naive approach that simply computes SSIM by regarding the whole feature as only one sub-feature: $\bar{\mathcal{S}} = \mathcal{S}(\boldsymbol{x},\boldsymbol{y})$.

2. MATRIX: From the viewpoint that the features are extracted from the spatial domain, the whole feature can be reshaped into a two-dimensional matrix of $I \times JK$ [11]. In this structure, we define the sub-features along the respective dimensions; $\hat{\boldsymbol{x}}_i = \{x_{ijk}\}_{j=1,k=1}^{J,K} \in \mathbb{R}^{JK}$, $\hat{\boldsymbol{x}}_{jk} = \{x_{ijk}\}_{i=1}^{I} \in \mathbb{R}^{I}$. The similarity measure is accordingly formulated as

$$\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{I} \frac{\mathcal{S}(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i)}{2I} + \sum_{j,k=1}^{J,K} \frac{\mathcal{S}(\hat{\boldsymbol{x}}_{jk}, \hat{\boldsymbol{y}}_{jk})}{2JK}. \tag{6}$$

Note that each feature element x_{ijk} is counted twice in this similarity measurement.

3. TENSOR: We treat the essential feature structure of three-way tensor as it is. The sub-features are consequently formulated along the respective three dimensions; $\hat{\boldsymbol{x}}_{ij} = \{x_{ijk}\}_{k=1}^{K} \in \mathbb{R}^{K}$, $\hat{\boldsymbol{x}}_{jk} = \{x_{ijk}\}_{i=1}^{I} \in \mathbb{R}^{I}$, $\hat{\boldsymbol{x}}_{ik} = \{x_{ijk}\}_{j=1}^{J} \in \mathbb{R}^{J}$. The feature elements in each sub-feature are consistent along one dimension. The similarity measure is given by

$$\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y}) = \tag{7}$$
$$\sum_{i,j=1}^{I,J} \frac{\mathcal{S}(\hat{\boldsymbol{x}}_{ij}, \hat{\boldsymbol{y}}_{ij})}{3IJ} + \sum_{j,k=1}^{J,K} \frac{\mathcal{S}(\hat{\boldsymbol{x}}_{jk}, \hat{\boldsymbol{y}}_{jk})}{3JK} + \sum_{i,k=1}^{I,K} \frac{\mathcal{S}(\hat{\boldsymbol{x}}_{ik}, \hat{\boldsymbol{y}}_{ik})}{3IK},$$

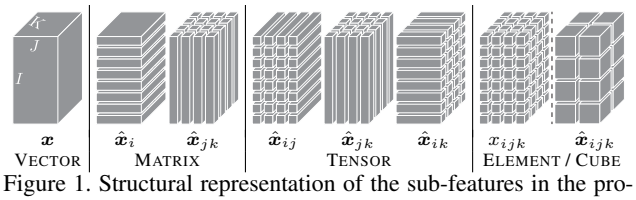

Figure 1. Structural representation of the sub-features in the proposed method. Each block indicates the sub-feature.

Table 1. Comparison of the sub-feature structures in terms of robustness, showing the ratio of the sub-features affected by one-element perturbation. The smaller ratio means higher robustness.

Structure	Ratio	Robustness rank
VECTOR	$\frac{1}{1}$	4th
MATRIX	$\frac{2}{I+JK}$	3rd
TENSOR	$\frac{3}{IJ+JK+KI}$	2nd
ELEMENT	$\frac{1}{IJK}$	1st

where each feature element x_{ijk} is counted three times.

4. ELEMENT: At the minimum case, we set each feature element x_{ijk} as the sub-feature, resulting in the simple similarity measure of

$$\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i,j,k=1}^{I,J,K} \frac{\mathcal{S}(x_{ijk}, y_{ijk})}{IJK} = \sum_{i,j,k=1}^{I,J,K} \frac{\mathcal{M}(\mathfrak{u}(x_{ijk}), \mathfrak{u}(y_{ijk}))}{IJK}$$
$$= \sum_{i,j,k=1}^{I,J,K} \frac{1}{IJK} \frac{2x_{ijk}y_{ijk}}{x_{ijk}^2 + y_{ijk}^2}. \tag{8}$$

where $\mathcal{V}$ and $\mathcal{C}$ are removed since the sub-feature is a scalar. This similarity measurement (8) is closely related to χ^2 distance $\sum_{i,j,k} \frac{1}{2} \frac{(x_{ijk}-y_{ijk})^2}{x_{ijk}+y_{ijk}}$, ignoring cross-bin relationships. And, as in the VECTOR structure, the ELEMENT approach does not take into account the structure of the feature at all. It is also possible to extend ELEMENT to CUBE by replacing point-wise element with a cube of $V \times V \times V$ volume; $\hat{\boldsymbol{x}}_{ijk} = \{x_{i'j'k'}\}_{i \le i' < i+V,\, j \le j' < j+V,\, k \le k' < k+V} \in \mathbb{R}^{V^3}$. The similarity measure is formulated in a manner similar to sliding window approach by

$$\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i,j,k=1}^{\substack{I-V+1,\\ J-V+1,\\ K-V+1}} \frac{\mathcal{S}(\hat{\boldsymbol{x}}_{ijk}, \hat{\boldsymbol{y}}_{ijk})}{(I-V+1)(J-V+1)(K-V+1)}. \tag{9}$$

Discussion. We can characterize these approaches from the viewpoint of robustness. Suppose one feature element is changed such as due to noise. The proposed similarity measure (5) is based on an ensemble of sub-features. Thus, degree of the effect by the one-element perturbation can be estimated as the number (ratio) of the sub-feature stained by it. This is summarized in Table 1. On the assumption that the local feature dimensionality I is generally larger than the numbers of spatial bins J and K, the above four approaches are ranked in terms of the robustness (ratio)

as VECTOR<MATRIX<TENSOR<ELEMENT. By considering that the ELEMENT approach (8) lacks structural information, the TENSOR one (7) is expected to work better.

3.2. Similarity measure

The original SSIM (1) is defined as the product of the three types of similarity function regarding mean $\mathcal{M}$, standard deviation $\mathcal{V}$ and correlation $\mathcal{C}$. The joint product is sensitive to any distortion of these statistics, which is favorable for image assessment but lacks robustness in feature matching. From the perspective of robustness, we have the following variants of SSIM measurement for $\mathcal{S}$ in (5):

$$\mathcal{S}_{org} = \mathcal{M}\times\mathcal{V}\times\mathcal{C} \quad \text{(original)} \tag{10}$$
$$\mathcal{S}_{+\mu} = w_{\mathcal{M}}\mathcal{M} + w_{\mathcal{C}}(\mathcal{V}\times\mathcal{C}) \quad \text{(separating } \mathcal{M}\text{)} \tag{11}$$
$$\mathcal{S}_{+\sigma} = w_{\mathcal{V}}\mathcal{V} + w_{\mathcal{C}}(\mathcal{M}\times\mathcal{C}) \quad \text{(separating } \mathcal{V}\text{)} \tag{12}$$
$$\mathcal{S}_{+c} = w_{\mathcal{C}}\mathcal{C} + w_{\mathcal{M}}(\mathcal{M}\times\mathcal{V}) \quad \text{(separating } \mathcal{C}\text{)} \tag{13}$$
$$\mathcal{S}_{add} = w_{\mathcal{M}}\mathcal{M} + w_{\mathcal{V}}\mathcal{V} + w_{\mathcal{C}}\mathcal{C} \quad \text{(fully additive)}, \tag{14}$$

where we introduce weights to balance the terms of additive forms. Note that (11) is the same configuration as the Euclidean one (4) by pushing out the similarity $\mathcal{M}$ of mean into the additive term.

Though the weights might be optimized by MKL [20], in this study, they are determined based on the value range of the similarity functions;

$$\mathcal{M}\in\begin{cases}[\,0,+1] : \text{non-neg. feat.}\\ [\text{-}1,+1] : \text{real feat.}\end{cases},\ \mathcal{V}\in[0,+1],\ \mathcal{C}\in[-1,+1], \tag{15}$$

where $\mathcal{M}$ takes a different range according to whether $\mathrm{u}(\boldsymbol{x}) \in [0,+\infty]$ or $[-\infty,+\infty]$. The weights can be set so as to make the similarity measures consistent in terms of value range. That is, in the case of non-negative features, $(w_{\mathcal{M}}, w_{\mathcal{V}}, w_{\mathcal{C}}) = (2,2,1)$, while for real-valued features, $(w_{\mathcal{M}}, w_{\mathcal{V}}, w_{\mathcal{C}}) = (1,2,1)$[1]. Note that those weights are finally normalized to ensure $\mathcal{S}(\hat{\boldsymbol{x}}_l, \hat{\boldsymbol{x}}_l) = 1$, resulting in $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{x}) = 1$ in (5); they are divided by $w_{\mathcal{M}} + w_{\mathcal{V}} + w_{\mathcal{C}}$.

Discussion. As to the robustness, if the perturbation appears independently in the three terms $\mathcal{M}, \mathcal{V}$ and $\mathcal{C}$, the fully additive form $\mathcal{S}_{add}$ (14) maximally suppresses the influence on the final similarity measure based on the similar discussion in Table 1. As a result, we recommend the fully additive similarity measurement $\mathcal{S}_{add}$ (14) in the TENSOR structure (7) which increases robustness by exploiting the additive formulation. Besides, the additive form has a merit of reducing dimensionality in explicit feature map (Sec. 3.3).

3.3. Explicit feature map

As in the previous methods [16, 13, 17], the proposed similarity measurement basically operates on pair-wise features $\{\boldsymbol{x},\boldsymbol{y}\}$ and, empirically speaking, such pair-wise operation requires significant computation time for plenty of samples. In contrast, L_2 metric can be efficiently computed by taking advantage of matrix multiplication such as via BLAS library. Especially for matching features, the fast computation of similarity measure is highly demanded. To reduce the computation time, we provide the explicit feature map $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^{D_g}$ such that $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y}) \approx \boldsymbol{g}(\boldsymbol{x})^\top \boldsymbol{g}(\boldsymbol{y})$ where the similarity computation results in simple matrix multiplication which is efficiently performed as in L_2 metric.

We first consider to decompose the similarity measurement function $\bar{\mathcal{S}}$ in a functional form.

Theorem 1. *For the proposed method of any similarity measure (10-14) under any structure (6-9), there exists the explicit functional map* $\mathrm{g}(\lambda;\boldsymbol{x})$ *such that* $\bar{\mathcal{S}}(\boldsymbol{x},\boldsymbol{y}) = \int_{-\infty}^{\infty} \mathrm{g}(\lambda;\boldsymbol{x})^* \mathrm{g}(\lambda;\boldsymbol{y}) d\lambda$.

The proposed similarity $\bar{\mathcal{S}}$ (5) of any configuration (Sec. 3.1, 3.2) is composed of addition and/or multiplication of $\mathcal{M}, \mathcal{V}$ and $\mathcal{C}$[2]. And, $\mathcal{C}(\boldsymbol{x},\boldsymbol{y})$ (3) is the dot product of the vectors $\boldsymbol{g}_{\mathcal{C}}(\boldsymbol{x}) = \frac{\boldsymbol{x}-\mathrm{u}(\boldsymbol{x})}{\|\boldsymbol{x}-\mathrm{u}(\boldsymbol{x})\|_2}$ and $\boldsymbol{g}_{\mathcal{C}}(\boldsymbol{y}) = \frac{\boldsymbol{y}-\mathrm{u}(\boldsymbol{y})}{\|\boldsymbol{y}-\mathrm{u}(\boldsymbol{y})\|_2}$. Therefore, the only issue for proving Theorem 1 is to prove that $\mathtt{k}(a,b) = \frac{2ab}{a^2+b^2}$ used in $\mathcal{M}$ and $\mathcal{V}$ has the explicit functional map.

Lemma 2. *There exists the explicit functional map* $\mathrm{g}_k(\lambda;a) = \tilde{\mathrm{g}}_k(\lambda;a) \oplus \mathtt{b}(a)$, *where* $\mathtt{b}(a) \in \mathbb{R}$, *such that* $\mathtt{k}(a,b) = \int_{-\infty}^{\infty} \tilde{\mathrm{g}}_k(\lambda;a)^* \tilde{\mathrm{g}}_k(\lambda;b) d\lambda + \mathtt{b}(a)\mathtt{b}(b)$.

Proof. We show the concrete form of g_k by following the approach [25] of the explicit map for χ^2 kernel.

In the case of $ab \neq 0$,

$$\mathtt{k}(a,b) = \frac{2\mathrm{sgn}(ab)}{\left|\frac{a}{b}\right| + \left|\frac{b}{a}\right|} = \frac{2\mathrm{sgn}(ab)}{e^{-\omega}+e^{\omega}} = \mathrm{sgn}(ab)\mathrm{sech}(\omega), \tag{16}$$

where $\omega = \log\left|\frac{b}{a}\right|$ and $\mathrm{sgn}(\cdot)$ is the sign function. Based on the Fourier expansion of sech, $\mathtt{k}$ is further rewritten as

$$\mathtt{k}(a,b) = \mathrm{sgn}(ab)\mathrm{sech}(\omega) = \mathrm{sgn}(ab)\int_{-\infty}^{\infty} e^{-i\omega\lambda}\kappa(\lambda)d\lambda$$
$$= \int_{-\infty}^{\infty} [\mathrm{sgn}(a)e^{-i\lambda\log|a|}\sqrt{\kappa(\lambda)}]^* [\mathrm{sgn}(b)e^{-i\lambda\log|b|}\sqrt{\kappa(\lambda)}]d\lambda, \tag{17}$$

where $\kappa(\lambda)$ is the inverse Fourier transform of $\mathrm{sech}(\omega)$, $\kappa(\lambda) = \frac{1}{2}\mathrm{sech}(\frac{\pi\lambda}{2})$. In the case of $ab = 0$, $\frac{2ab}{a^2+b^2} = [\![a = 0]\!][\![b = 0]\!]$ where $[\![\cdot]\!]$ is the Iverson bracket that equals to 1 if the condition in the brackets is satisfied and 0 otherwise.

[1] In real-valued features, $\mathcal{M}, 2\mathcal{V}-1$ and $\mathcal{C}$ have the identical value rage of $[-1,+1]$, and the constant bias in $2\mathcal{V}-1$ is inessential and removed.

[2] In the explicit mapping, $+$ and $\times$ in (5, 10-14) are replaced with $\oplus$ (direct sum) and $\otimes$ (direct product), respectively. And note that in the mapping the square root is applied to the weights $w_l, w_{\mathcal{M}}, w_{\mathcal{V}}, w_{\mathcal{C}}$.

Therefore, we can obtain

$$\tilde{\mathrm{g}}_k(\lambda; a) = \mathrm{sgn}(a) e^{-i\lambda \log |a|} \sqrt{\frac{1}{2}\mathrm{sech}\left(\frac{\pi\lambda}{2}\right)}, \quad (18)$$

$$\mathtt{b}(a) = [\![a = 0]\!]. \qquad \square$$

By using Lemma 2, we can give the explicit functional maps of $\mathcal{M}$ and $\mathcal{V}$ to finally prove Theorem 1. And, the fixed dimensional explicit feature map $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^{D_g}$ is obtained through approximating the function $\mathrm{g}_k(\lambda; a)$ in a vector form. According to [25], $\tilde{\mathrm{g}}_k(\lambda; a)$ is approximated by means of $\tilde{D}_k$ basis points in λ, resulting in $\tilde{D}_k$-dimensional vector, of which direct sum with $\mathtt{b}(a)$ forms $D_k = \tilde{D}_k + 1$-dimensional vector $\boldsymbol{g}_k(a)$. The explicit maps of $\mathcal{M}$ and $\mathcal{V}$ are thus simply obtained by $\boldsymbol{g}_k(\mathtt{u}(\hat{\boldsymbol{x}}_l))$ and $\boldsymbol{g}_k(\mathtt{q}(\hat{\boldsymbol{x}}_l))$, respectively, and as presented above, $\mathcal{C}$ has the explicit map $\boldsymbol{g}_{\mathcal{C}}(\hat{\boldsymbol{x}}_l) = \frac{\hat{\boldsymbol{x}}_l - \mathtt{u}(\hat{\boldsymbol{x}}_l)}{\|\hat{\boldsymbol{x}}_l - \mathtt{u}(\hat{\boldsymbol{x}}_l)\|_2}$ of which dimensionality is the same as that of the sub-feature $\hat{\boldsymbol{x}}_l$. For example, the dimensionality of the explicit map $\boldsymbol{g}(\boldsymbol{x})$ of the proposed similarity, S_{add} in TENSOR structure, is $3IJK + 2(\tilde{D}_k + 1)(IJ + JK + KI)$ where $\tilde{D}_k$ is the only parameter [25].

The explicit feature map is not only useful for speeding up similarity measurement but also regarded as a novel feature transform. Thereby, the proposed method works for feature matching as well as feature *classification* using the feature map $\boldsymbol{g}(\boldsymbol{x})$; a *linear* classifier such as by SVM [24] is applicable to the feature vectors $\boldsymbol{g}(\boldsymbol{x})$ in which the proposed similarity measure is embedded.

3.4. Metric property

A *metric* property inheres in the proposed similarity $\bar{\mathcal{S}}$.

Theorem 3. *$\sqrt{1 - \bar{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{y})}$ is a metric.*

Proof. The proposed method of any configuration is ensured to have $\bar{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{x}) = 1,\ \forall \boldsymbol{x}$. And, we apply Theorem 1 to obtain

$$1 - \bar{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}(\bar{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{x}) + \bar{\mathcal{S}}(\boldsymbol{y}, \boldsymbol{y}) - 2\bar{\mathcal{S}}(\boldsymbol{x}, \boldsymbol{y}))$$

$$= \frac{1}{2}\int_{-\infty}^{\infty} \mathrm{g}(\lambda;\boldsymbol{x})^* \mathrm{g}(\lambda;\boldsymbol{x}) + \mathrm{g}(\lambda;\boldsymbol{y})^* \mathrm{g}(\lambda;\boldsymbol{y}) - 2\mathrm{g}(\lambda;\boldsymbol{x})^* \mathrm{g}(\lambda;\boldsymbol{y}) d\lambda$$

$$= \frac{1}{2}\langle \mathrm{g}(\lambda;\boldsymbol{x}) - \mathrm{g}(\lambda;\boldsymbol{y}), \mathrm{g}(\lambda;\boldsymbol{x}) - \mathrm{g}(\lambda;\boldsymbol{y}) \rangle. \quad (19)$$

The square root of this measure is a metric. $\square$

The property of SSIM regarding a metric is partially mentioned in [3]. The metric property would be useful for more efficient data structures and search algorithms.

4. Experiment

The proposed similarity measurement is basically useful for matching structured features (Sec. 4.1, Sec. 4.2). In addition, via the explicit feature map in Sec. 3.3, the method is also applicable to feature classification tasks (Sec. 4.3).

4.1. Keypoint matching

We first test the proposed method on the task of keypoint matching by means of local descriptors. The local descriptors are generally formulated in a structured tensor form exploiting local spatial layout, such as $8_{(I)} \times 4_{(J)} \times 4_{(K)}$ for SIFT [15] and SURF [1]. Performance for the matching is evaluated on the dataset by Mikolajczyk and Schmid [8] in a similar protocol. The dataset contains eight image sets each of which consists of one reference (undistorted) image and five distorted ones captured at different angle, scale and so on; in total, there are 40 image pairs for evaluating local descriptor matching. In this evaluation, we extract SIFT [15] local descriptors $\boldsymbol{x} \in \mathbb{R}^{8\times4\times4}$ on the keypoints detected by a Hessian-based detector. Since we focus only on evaluating (dis)similarity measure, the performance is measured based on averaged precision (AP), the ratio of the correctly matched descriptor pairs (of $>60\%$ overlap).

Feature structure. We evaluate various types of sub-feature structure (Sec. 3.1 and Fig. 1) with fixing the similarity measure $\mathcal{S} = \mathcal{S}_{org}$ (10). The performance is compared on the basis of TENSOR structure which is of our main interest. As shown in Fig. 2, the TENSOR structure is superior to the other types of structure; in particular, it significantly outperforms the VECTOR structure. Actually, the VECTOR structure which simply applies SSIM to feature vectors is inferior even to the standard L_2 metric (Fig. 3). The MATRIX structure performs relatively well, though being still inferior to the TENSOR one, and both the structures surpass the VECTOR and ELEMENT ones which do not take into account the structure of SIFT feature at all. This result demonstrates effectiveness of incorporating the feature structure into similarity measurement. Although the CUBE structure slightly exploits such structure characteristics, it is necessary to form consistent sub-features for similarity measurement; in the TENSOR structure, the sub-features are consistent along respective dimensions, while CUBE one mixes up all the three dimensions in the sub-features.

To further demonstrate the effectiveness to incorporate intrinsic SIFT structure, we additionally tested the method that randomly permutes feature elements in the identical TENSOR structure. The random permutation of feature elements largely degrades inherent physical meaning of the SIFT structure, harming consistency in the sub-feature, and accordingly pollutes the performance as shown in Fig. 2. This experimental result shows that it is important to deal with the intrinsic feature structure as it is.

Similarity measure. Next, we go into the similarity measurement $\mathcal{S}$ used in the TENSOR structure. Various types of similarity measurement functions (10-14) are compared with $\mathcal{S}_{add}$ of our main interest. Fig. 4 shows that $\mathcal{S}_{add}$ is superior to $\mathcal{S}_{org}$ while producing comparable performance with the other methods based on additive forms ($\mathcal{S}_{+\mu}, \mathcal{S}_{+\sigma}, \mathcal{S}_{+c}$). Unfolding the original SSIM formulation

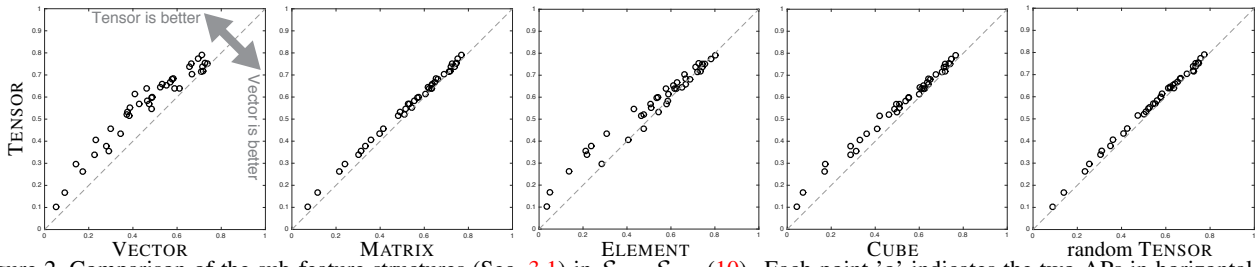

Figure 2. Comparison of the sub-feature structures (Sec. 3.1) in $\mathcal{S} = \mathcal{S}_{org}$ (10). Each point 'o' indicates the two APs in horizontal and vertical axes, produced by compared two methods for each image pair. All the vertical axes indicate TENSOR structure.

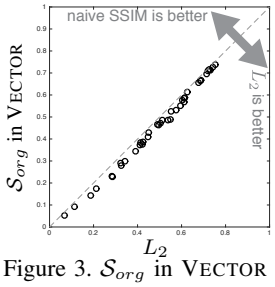

Figure 3. $\mathcal{S}_{org}$ in VECTOR (naive SSIM) vs. L_2.

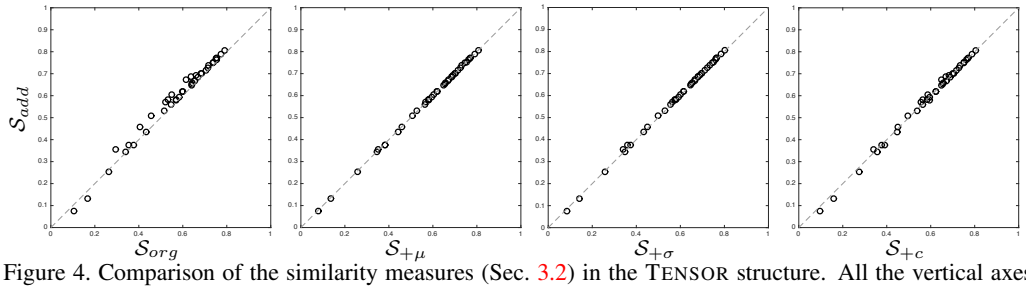

Figure 4. Comparison of the similarity measures (Sec. 3.2) in the TENSOR structure. All the vertical axes indicate $\mathcal{S}_{add}$ (14).

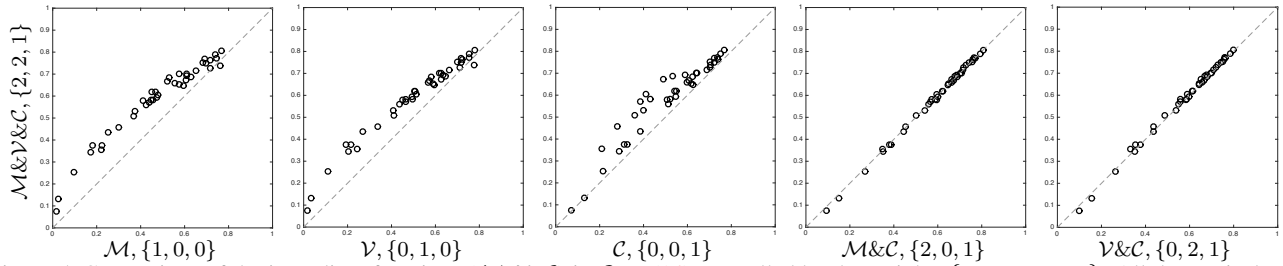

Figure 5. Comparison of the ingredient functions $(\mathcal{M}, \mathcal{V}, \mathcal{C})$ in $\mathcal{S}_{add}$ (14) controlled by the weights $\{w_{\mathcal{M}}, w_{\mathcal{V}}, w_{\mathcal{C}}\}$. All the vertical axes indicate $\{w_{\mathcal{M}}, w_{\mathcal{V}}, w_{\mathcal{C}}\} = \{2, 2, 1\}$.

$\mathcal{S}_{org}$ into additive forms improves the performance with increasing the robustness. Among the additive forms (11-14), $\mathcal{S}_{add}$ is preferable from the viewpoint of computation efficiency since it produces the smallest dimensionality of the explicit feature map (Sec. 3.3).

In $\mathcal{S}_{add}$, we further investigate the roles of the three ingredient functions $\mathcal{M}, \mathcal{V}$ and $\mathcal{C}$ through controlling the weights $\{w_{\mathcal{M}}, w_{\mathcal{V}}, w_{\mathcal{C}}\}$, which were set to $\{2, 2, 1\}$ in the above experiment as described in Sec. 3.2. As shown in Fig. 5, any single function poorly works while the performance is significantly improved by combining two of them, being comparable to the full combination with $\{w_{\mathcal{M}}, w_{\mathcal{V}}, w_{\mathcal{C}}\} = \{2, 2, 1\}$. For the computation efficiency, it is preferable to construct the similarity measurement $\mathcal{S}_{add}$ by using less number of ingredients for reducing the dimensionality of the explicit feature map (Sec. 3.3). Thus, we employ simpler configuration of $\mathcal{S}_{add}$ with $w_{\mathcal{M}} = 0$ or $w_{\mathcal{V}} = 0$; the dimensionality of those explicit feature maps is $3IJK + (\tilde{D}_k + 1)(IJ + JK + KI)$, and in this case of SIFT descriptor, it results in 1024-dimensional feature vector by $\tilde{D}_k = 7$. More practically speaking, the method composed only of $\mathcal{V}$ and $\mathcal{C}$ is favorable since we can use fixed weights of $\{w_{\mathcal{M}}, w_{\mathcal{V}}, w_{\mathcal{C}}\} = \{0, 2, 1\}$ regardless of feature domain (non-negative or real-valued).

As a conclusion, from perspectives of performance and practical use, it is advantageous to employ the proposed method of $\mathcal{S}_{add}$ with $\{w_{\mathcal{M}}, w_{\mathcal{V}}, w_{\mathcal{C}}\} = \{0, 2, 1\}$ in the TENSOR structure, which is thus applied in what follows.

4.1.1 Comparison with the other methods

Then, the proposed method is compared to the other methods of distance (similarity) measurement including the standard L_2 metric, χ^2 distance [2], diffusion distance (DiffuseDist) [13], SIFT distance (SiftDist) [16] and fast Earth Mover's Distance (fEMD) [17]. The performance results are shown in Fig. 6 on the basis of the proposed method which favorably outperforms the others. This result demonstrates that (1) the feature structure (tensor) in the proposed method is a favorable standpoint than a histogram form imposed on the previous methods except for L_2 metric, and (2)

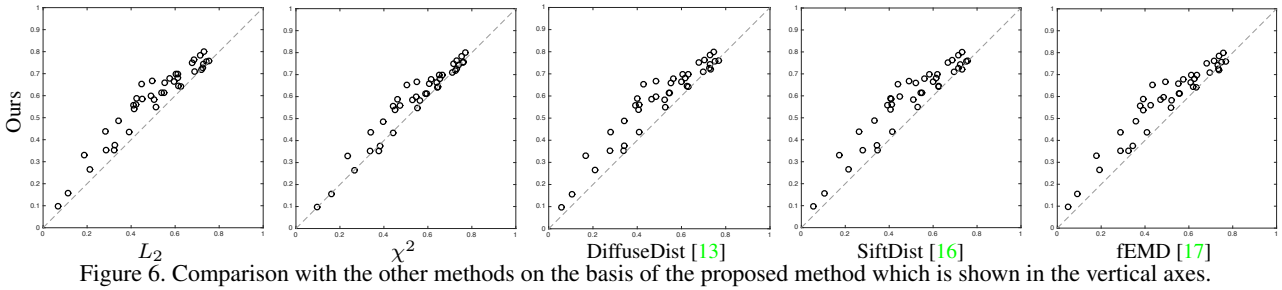

Figure 6. Comparison with the other methods on the basis of the proposed method which is shown in the vertical axes.

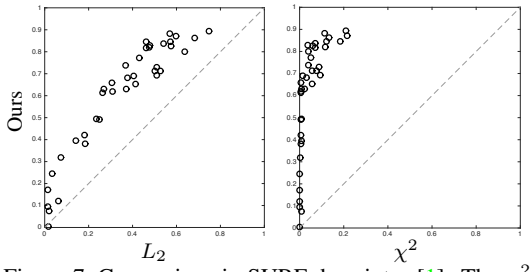

Figure 7. Comparison in SURF descriptor [1]. The χ^2 distance is applied by force just for a reference, though it is unsuitable to this type of feature.

Table 2. Computation time per sample pair in $M = N = 4096$ samples.

Method	Time (nsec)
L_2	15.2
χ^2	569.0
DiffuseDist [13]	17898.6
SiftDist [16]	1260.3
fEMD [17]	2105396.0
Ours w/o map	1146.5
Ours with map	65.7

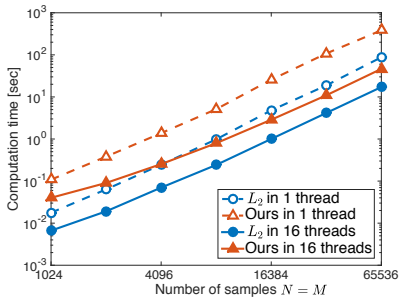

Figure 8. Computation time in multi-threading.

the SSIM-based measure is suitable for extracting cross-bin similarity compared to diffusion distance [13].

In addition to SIFT descriptors, we apply the proposed similarity metric to the SURF detector&descriptor (SURF-128) [1]. The SURF descriptor characterizes a local image region by means of 8-dimensional gradient filter responses on 4×4 spatial grids, allowing negative feature values. Thereby, it produces the structured feature $\boldsymbol{x} \in \mathbb{R}^{8\times4\times4}$ which is the same three-way tensor structure as in SIFT but defined in a real-valued feature, not a (non-negative) histogram. It is unsuitable to apply the previous methods [2, 13, 16, 17] based on a histogram form to this type of descriptor; for example, the χ^2 distance obviously degrades performance as shown in Fig. 7. The proposed method significantly outperforms the L_2 metric.

We also show the computation time in measuring the similarity between sets of M and N samples (MN pairs to be compared). The proposed method is composed of two steps of (1) computing the explicit feature map of the input features in $O((M+N)D)$ and then (2) performing matrix multiplication in $O(MND_g)$ where $D = IJK$ is the input feature dimensionality. Therein, the first step of feature mapping is negligible and the second one of matrix multiplication dominates the computation time, which can be efficiently performed such as by BLAS library. Table 2 shows the computation time[3] per sample pair and we can see that the proposed method is significantly speeded up by the explicit feature map (Sec. 3.3) and is much faster than the other methods except for L_2 metric; this is, of course, due to the dimensionality of the features which corresponds to D in L_2 and $\{3+D_k(\frac{1}{I}+\frac{1}{J}+\frac{1}{K})\}D$ in the proposed method. In addition, the method can be easily parallelized such as by applying multi-thread BLAS as shown in Fig. 8. As a result, we can say that the proposed method achieves high performance in fast computation time for feature matching.

4.1.2 Feature matching vs. image assessment

As an aside, we mention the (in)applicability of the method to image quality assessment which is the main target of the original SSIM, though such task is out of our focus. The proposed method can produce similarity between *images* in a manner similar to SSIM (Sec. 2). On the TID2008 dataset [19], the method produces the evaluation score[4] of 0.5489 which is slightly inferior to that of the original SSIM, 0.5768. This result contrasts with the above experimental results of feature matching, due to the different objectives of those tasks. As described in Sec. 3, it is necessary for image assessment to extract detailed difference (distortion) affecting human perception, while feature matching demands high robustness to inessential difference with extracting discriminativity of the targets. The proposed method (Sec. 3) is carefully constructed for enhancing robustness according to the objective of feature matching.

4.2. Image retrieval

The proposed method is then tested on an image retrieval task which picks up similar images based on the descrip-

[3]The methods are implemented in MATLAB mex-C, and the computation time is measured on Xeon 3.4GHz PC for $\boldsymbol{x} \in \mathbb{R}^{8\times4\times4}$.

[4]The performance is measure based on Kendall's rank correlation coefficient between the estimated similarity and manually annotated one.

Table 3. Image retrieval performance on Oxford building dataset [18] in the framework of [7]. Our method consists of $\mathcal{S}_{add}$ with $\{w_{\mathcal{M}}, w_{\mathcal{V}}, w_{\mathcal{C}}\} = \{0, 2, 1\}$ in TENSOR structure.

Method	L_2	χ^2	$\mathcal{S}_{org}$ in MATRIX	$\mathcal{S}_{org}$ in TENSOR	Ours
mAP	67.6	71.2	70.3	72.0	**73.2**

tor matching in the framework of [7]. This task is different from keypoint matching (Sec. 4.1) in that the similarity is measured between different images of different targets while in Sec. 4.1 the reference and distorted images of the identical target are compared requiring high robustness to the distortion. Therefore, in this task, we can evaluate the similarity measure in terms of discriminating objects.

The SIFT descriptors are extracted on the keypoints detected by DoG detector for retrieval and we compare the methods on similarity (distance) measurement used in k-NN search; for detailed pipeline of the image search, refer to [7]. The retrieval performance on the Oxford building dataset [18] is shown in Table 3. We can see that the TENSOR structure is superior to the MATRIX one and the measure $\mathcal{S}_{add}$ (additive) is preferable compared to $\mathcal{S}_{org}$ (multiplicative) as is the case with the keypoint matching (Sec. 4.1). The proposed method outperforms L_2 metric with a large margin and χ^2. This result demonstrates that the proposed similarity measure is effective for discriminating object (parts).

4.3. Image classification

Lastly, the proposed method is evaluated in a framework of classification which is slightly different from feature matching. It is possible to employ the pair-wise similarity for classification via exemplar-based classifiers such as k-NN, but it requires substantial computation cost. The proposed method is capable of transforming the input features via the explicit feature map to a vector form in which the similarity metric is embedded, and an efficient linear classifier by SVM [24] can be directly applied.

The method is first applied to transform HOG features on a person classification task using INRIA person dataset [4]. The HOG feature extracted from an image patch of 64×128 pixels by the method [6] is formulated in a three-way TENSOR structure of $\mathbb{R}^{31\times8\times16}$, where cells of 8×8 pixels are employed to produce 8×16 grids. The performance results are shown in Fig. 9 demonstrating that the proposed method effectively improves the performance of the original HOG feature. And, it also outperforms the χ^2-based method [25] that gives an explicit feature map of χ^2 kernel; note that the proposed method produces smaller dimensional feature vector ($\mathbb{R}^{18880}$) than the method [25] ($\mathbb{R}^{27776}$) with $\tilde{D}_k = 7$.

Then, the method is also applied to CNN features [5] on scene classification tasks using Scene-15 [12] and SUN-397 [30] datasets. CNN image features have been applied to various classification tasks with great success and in this experiment we employ the very deep CNN model [23] trained on ImageNet dataset and extract the features as in [5] by using `pool5` layer to produce 512×8×8 TENSOR feature from an image of 256×256 pixels in scene-15 and 512×12×12 feature from an image of 384×384 pixels in SUN-397[5]. Scene images contain layout of scene parts [12] which is parsed by the spatial grid features. As shown in Table 4, the proposed method favorably improves the performance of the original CNN feature and is competitive with the state-of-the-art method [32] which employs CNN features trained on a large *scene* dataset.

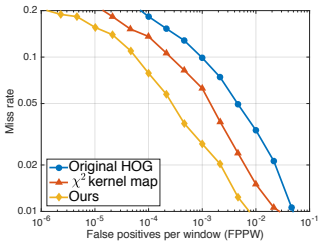

Figure 9. Comparison in transforming HOG feature [6] on INRIA dataset [4].

Table 4. Scene classification performance (%).

(a) Scene-15 dataset [12]

Method	Acc.
Kobayashi [9]	85.6±0.7
Places-CNN [32]	90.2±0.3
original CNN-feat.	90.3±0.2
Ours	**91.2**±0.5

(b) SUN-397 dataset [30]

Method	Acc.
Sánchez *et al.* [22]	47.2±0.2
Places-CNN [32]	54.3±0.1
original CNN-feat.	52.1±0.3
Ours	**54.3**±0.3

These experimental results show that the proposed method works as measuring feature similarity for matching as well as transforming features to improve classification performance.

5. Conclusion

We have proposed a novel method to measure a similarity metric between pairs of structured features. The proposed method leverages SSIM [27], which has been successfully applied in image quality assessment, to construct similarity measure of high robustness by effectively exploiting an intrinsic structure of the features, such as three-way *tensor*. In addition, we provide the explicit feature map such that the proposed similarity metric is embedded as a dot product, in order to significantly speed up the similarity measurement as well as to transform the feature into an effective vector form which is directly fed into linear classifiers. The experimental results on various tasks demonstrate the effectiveness of the proposed method from both aspects of feature matching and classification.

[5]The `pool5` layer [23] produces 512-dimensional feature on the receptive field of 32×32 pixels, and images in scene-15 dataset are resized to 256×256 pixels according to the average image size, while in SUN-397 images are resized to 384×384 due to the larger average size of images.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.

[3] D. Brunet, E. R. Vrscay, and Z. Wang. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4):1488–1499, 2012.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[7] H. Jégou, M. Douze, and C. Schmid. Exploiting descriptor distances for precise image search. Rapport de recherche RR-7656, INRIA, June 2011.

[8] k. Mikolajczyk and c. Schmid. A performance evaluation of local descriptors. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[9] T. Kobayashi. Bof meets hog: Feature extraction based on histograms of oriented p.d.f gradients for image classification. In *CVPR*, pages 747–754, 2013.

[10] T. Kobayashi. Dirichlet-based histogram feature transform for image classification. In *CVPR*, pages 3278–3285, 2014.

[11] T. Kobayashi. Low-rank bilinear classification: Efficient convex optimization and extensions. *International Journal of Computer Vision*, 110(3):308–327, 2014.

[12] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[13] H. Ling and K. Okada. Diffusion distance for histogram comparison. In *CVPR*, 2006.

[14] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*, pages 28–42, 2008.

[15] D. Lowe. Distinctive image features from scale invariant features. *International Journal of Computer Vision*, 60:91–110, 2004.

[16] O. Pele and M. Werman. A linear time histogram metric for improved sift matching. In *ECCV*, pages 495–508, 2008.

[17] O. Pele and M. Werman. Fast and robust earth mover's distances. In *ICCV*, 2009.

[18] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *ICCV*, 2007.

[19] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti. TID2008 - a database for evaluation of full-reference visual quality assessment metrics. *Advances of Modern Radioelectronics*, 10:30–45, 2009.

[20] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[21] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[22] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[24] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[25] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, 2010.

[26] Z. Wang and A. C. Bovik. Mean squared error: love it or leave it? - a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, January 2009.

[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactios on Image Processing*, 13(4):600–612, April 2004.

[28] Z. Wang and E. P. Simoncelli. Translation insensitive image similarity in complex wavelet domain. In *ICASSP*, pages 573–576, 2005.

[29] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multi-scale structural similarity for image quality assessment. In *IEEE Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1398–1402, 2003.

[30] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[31] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

[32] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495, 2014.