International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 14 (2017) pp. 4130-4137

Object Recognition Using Log-Euclidean Multivariate Gaussian Descriptors

B. Ramesh Naik[1] and T. Venu Gopal[2]

[1]*Assistant Professor, Department of CSE, GST, GITAM University, Bengaluru, India*

[2]*Professor, Department of CSE, JNTUH College of Engineering, Sulthanpura, Hyderabad, India.*

[1]Orcid: 0000-0002-9082-7364

Abstract

Visual features such as color, texture and shape are used extensively to represent image signatures in content based image retrieval systems (CBIR). The retrieval quality is adequate for some retrieval tasks but there is still a semantic gap between the low-level visual features (textures, shapes, colors) and automatically extracted high-level concepts that users normally search for. We proposed novel approach based on Log-Euclidean Multivariate Gaussian to represent the image visual features efficiently which enjoy the important applications of computer vision. This approach extracts image features using SIFT algorithm and these features again represented using multivariate Gaussian. The resulting descriptors called Log-Euclidean Multivariate Gaussian Descriptors, works well with low and high dimensional raw features. Extensive experiments were conducted to evaluate thoroughly this approach and the result showed that this approach is very competitive in object recognition and classification.

Keywords: CBIR, SIFT, Covariance Matrices, Multivariate Gaussian Descriptors, Gaussian Space, Object Recognition and Classification.

INTRODUCTION

With the advent of the rapid growth in Technology and World Wide Web (WWW), large volume of image databases are now made available over the web for information sharing and there is immediate need to efficiently organize this data and retrieve the information from such large volumes of image databases. As large image databases are being built, organizing the image databases for efficient retrieval is a non-trivial problem. General visual features of image are color, texture, shape, spatial relationship. These low level visual features are used extensively to represent image signatures. Visual features can be very general or domain specific. Visual content is application dependent and may involve domain knowledge. Content-based visual information retrieval (CBVIR) and Content-based image retrieval (CBIR) [1], [2] are the applications of computer vision to the image retrieval problem that is, searching for digital images in large databases using visual features such as color, texture and shape. In CBIR [1], [2] searching the images based on their visual contents, instead of relying on human-inputted metadata such as captions or keywords. Though many sophisticated Image representation algorithms such as Fourier Transforms, Wavelet Transforms, Generic moments been designed to extract visual features, these algorithms cannot adequately model image semantics and have many limitations when dealing with large volume of image Databases. The Image retrieval quality is sufficient for some retrieval tasks but there is still a semantic gap between the low-level visual features (textures, shapes, colors) and automatically extracted high-level concepts that users normally search for. More specifically, the 'semantic gap' is referred to as the discrepancy between the limited descriptive power of low-level visual features and the richness of user semantics. Comprehensive surveys exist on this topic of semantic gap in Content Based Image Retrieval. In particular, densely sampled descriptors have proven to achieve outstanding performance in image-based classification tasks such as object classification, texture recognition, scene categorization and image retrieval [25]. However, it is challenging to develop image descriptors with high distinctiveness for general image Retrieval System.

Our goal is to propose a new function valued descriptors that represent the image visual features efficiently which enjoy the important application of recognizing objects from given image database. The proposed approach is based on Multivariate Gaussian descriptor which takes input as SIFT features [9] and then describes new features using Log Euclidean Multivariate Gaussian. Representing image features in d-dimensional space N(d) using multivariate Gaussian(μ, $\sum$) [4], [5] is challenging task which is not a linear space but a manifold. The space of the Gaussian can be provided with lie group structure by defining multiplication operation on manifold [3], [15]. The process of embedding Gaussian in linear space is shown in detail in this paper.

Our work is inspired by popular distribution based descriptors such as SIFT [9], and HoG [21]. This approach initially extracts image features using SIFT [12]. These features are again described using Log Euclidean Multivariate Gaussian. These features are called as Log-Euclidean Multivariate Gaussian descriptors.

First we introduced the details of SIFT descriptors and Multivariate Gaussian Distribution. Next we described Gaussian Embedding in Linear Space and Log Euclidean Multivariate Gaussian Descriptors. Then we presented the experimental results to evaluate and analyze our Descriptors. Finally we have concluded the paper.

SCALE INVARIANT FEATURE TRANSFORM

Primary aspect of CBIR is image matching, which is important aspect of many problems of computer vision and pattern recognition communities. But it is also challenging vision problems because images often suffer from significant scale, illumination variations, material recognition, pose changes, background clutter, partial occlusion [7], [8]. The scale invariant feature transform (SIFT) [9], [11] is a feature extraction approach which generates high dimensional features from patches selected based on pixel values which can then be compared and matched to other features. This approach has a set of parameters, which can be varied and the choice, modification can be used to improve the quality of the results. In the original paper by David Lowe [9] a set of default parameters is given with a variety of images but whether or not these are optimal is not clear.

This section introduces basic background of SIFT. The original SIFT feature detection algorithm developed by David Lowe [9], [13] is a four stage process that creates unique and highly descriptive features from an image. The features extracted using SIFT are designed to be invariant to rotation, changes in scale, illumination, noise and small changes in viewpoint. The features can be used to indicate if there is any correspondence between areas within images. Clusters of features from an image that are similar to a cluster of features from another image may indicate, with a high likelihood, areas that match. This allows object recognition to be implemented by comparing features generated from input images to features generated from images of target objects. The four stages of the SIFT algorithm is given in Lowe's paper [9]. The four stages of the SIFT algorithm are as follows.

Scale-space Extrema Detection.

The first step to find the SIFT features is to create a Gaussian scale-space pyramid for the image. Multiple octaves are created from blurred images using convolution of Gaussian. Difference between two consecutive images within an octave is referred as Difference of Gaussian. The scale space is defined by the function:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Where * is the convolution operator, $G(x, y, \sigma)$ is a variable-scale Gaussian and $I(x, y)$ is the input image.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma} e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}}$$

Stable key point locations in the scale-space are detected by various techniques. Among which Difference of Gaussians is one such technique, locating scale-space extrema, $D(x, y, \sigma)$ by computing the difference between two images, one with scale k times the other. $D(x, y, \sigma)$ is then given by:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

To detect the local maxima and minima of $D(x, y, \sigma)$ each point is compared with its 8 neighbors at the same scale, and its 9 neighbors up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema.

Feature Localisation.

The next step in SIFT is Feature Localization. The number of features which are less important is reduced in this stage. Interpolation occurs to locate the exact, sub pixel, location of the candidate features and points that are in areas of low contrast or those that are localised along edges are eliminated. The location of extremum, $\mathbf{z}$ is given by:

$$\mathbf{z} = -\frac{\partial^2 D^{-1}}{\partial X^2} \quad \frac{\partial D}{\partial X}$$

If the function value at $\mathbf{z}$ is below a threshold value then this point is excluded. This removes extrema with low contrast. To eliminate extrema based on poor localization it is noted that in these cases there is a large principle curvature across the edge but a small curvature in the perpendicular direction in the difference of Gaussian function. If this difference is below the ratio of largest to smallest eigenvector, from the 2x2 Hessian matrix at the location and scale of the key point, the key point is rejected.

Orientation Assignment

The image gradient directions of the pixels in a feature's neighbourhood are calculated and added to an orientation histogram with 36 bins [13]. The values in the neighbourhood are Gaussian weighted so those nearer the centre have a greater effect on the resulting orientation. One key orientation is selected for each feature. The approach taken to find an orientation is as followed: Use the key points scale to select the Gaussian smoothed image L, from

above Gradient magnitude (m) is

$$m(x,y) = \sqrt{(L(x+1,y)-L(x-1,y))^2 + (L(x,y+1)-L(x,y-1))^2}$$

Orientation (θ) is

$$\theta(x,y) = \tan(L(x,y+a) - \frac{L(x,y-1)}{(L(x+1,y)-L(x-1,y))})$$

Form an orientation histogram from gradient orientations of sample points.

Find out the highest peak in the histogram. Use this peak and any other local peak within 80% of the height of this peak to create a keypoint with that orientation some points will be assigned multiple orientations. Fit a parabola to the 3 histogram values closest to each peak to interpolate the peaks position

Creating the Feature Descriptor

A Feature Descriptor is considered with 128 dimensional Vector which describes pixel properties of area surrounding a feature is shown in Fig-1. A 4 by 4 array of 16 histograms is centred on the feature and rotated to match the key orientation calculated in the previous step. The gradient magnitudes are given a Gaussian weighting, added to the histograms and normalised to create the descriptor.

The local gradient data, used above, is also used to create keypoint descriptors. The gradient information is rotated to line up with the orientation of the keypoint and then weighted by a Gaussian with variance of 1.5 * keypoint scale. This data is then used to create a set of histograms over a window centred on the keypoint.

Keypoint descriptors typically uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. These results in a feature vector containing 128 elements.

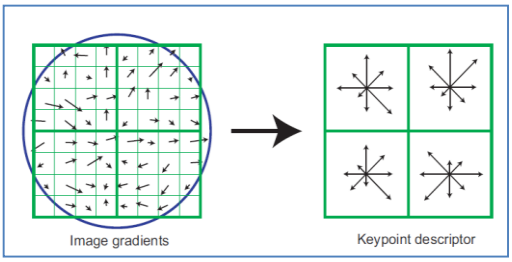

Figure 1: Image Gradients and Key Point Descriptors

MULTIVARIATE GAUSSIAN DESCRIPTOR

Let a d-dimensional random vector X = (X_1,X_2,,,,,X_d) , the multivariate Gaussian distribution[10], [17], [18] or normal distribution on R^d if there is a vector $\xi \in R^d$ and a dxd matrix ∑ such that

$$\lambda^T X \sim \mathcal{N}(\lambda^T\xi, \lambda^T\Sigma\lambda) \text{ for all } \lambda \in R^d \quad (1)$$

We then write X ~ $\mathcal{N}_d$ (ξ , ∑) Taking λ = ei or = ei + ej where ei is the unit vector with i-th coordinate 1 and the remaining equal to zero yields:

$$Xi \sim \mathcal{N}(\xi_i, \sigma_{ii}), \text{Cov}(Xi, Xj) = \sigma ij$$

Hence ξ is the mean vector and ∑ the covariance matrix of the distribution. The definition (1) makes sense if and only if $\lambda^T\sum\lambda \geq 0$, i.e. if ∑ is positive semi definite [6]. Note that we have allowed distributions with variance zero. The multivariate moment generating function of X can be calculated using the relation (1) i.e.

$$m_d(\lambda) = E\{e^{\lambda^T X}\} = e^{\lambda^T\xi + \lambda^T\Sigma\lambda/2} \quad (2)$$

In Eq-2, we have used that the univariate moment generating function for $\mathcal{N}$ (μ, σ^2) is

$$m_1(t) = e^{t\mu + \frac{\sigma^2 t^2}{2}} \text{ and let } t = 1, \quad \mu = \lambda^T\xi, \quad \text{and} \quad \sigma^2 = \lambda^T\Sigma\lambda$$

In particular this means that a multivariate Gaussian distribution is determined by its mean vector and covariance matrix.

Assume X^T = (X_1,X_2,X_3) with Xi independent and Xi ~ $\mathcal{N}$ (ξi , σ_i^2) Then

$$\lambda^T X = \lambda_1 X_1 + \lambda_2 X_2 + \lambda_3 X_3 \sim \mathcal{N}(\mu, r^2) \quad (3)$$

$$with\ \mu = \lambda^T\xi = \lambda_1\xi_1 + \lambda_2\xi_2 + \lambda_3\xi_3,$$

$$r^2 = \lambda_1^2\sigma_1^2 + \lambda_2^2\sigma_2^2 + \lambda_3^2\sigma_3^2$$

$$hence\ X \sim \mathcal{N}_3(\xi, \Sigma) \text{with } \xi^T = (\xi_1, \xi_2, \xi_3) \text{ and}$$

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix}$$

If ∑ is positive semi definite i.e $\lambda^T\Sigma\lambda > 0\ for\ \lambda \neq 0$ the distribution has density on R^d

$$f(X|\xi,\Sigma) = (2\pi)^{-\frac{d}{2}}(\det K)^{\frac{1}{2}} e^{-(x-\xi)^T K(X-\xi)/2} \quad (4)$$

In Eq-4, K = $\sum^{-1}$ is the concentration matrix of the distribution. We then also say that ∑ is regular if $X_1, \ldots . X_d$ are independent and $X_i \sim \mathcal{N}(\xi_i, \sigma_i^2)$ And their joint density has the form (2) with

$$\Sigma = \text{diag}(\sigma_i^2) \text{ and } K = \Sigma^{-1} = \text{diag}(1/\sigma_i^2) \quad (5)$$

Hence vectors of independent Gaussians are multivariate Gaussian [10], [18]. To match features often the Euclidean

distance between two feature vectors is used to find the nearest neighbour.

EMBEDDING GAUSSIAN IN LINEAR SPACE

Direct Embedding

Let us consider the set

$$S(n+1) = \left\{St, X \triangleq \begin{bmatrix} X & t \\ 0^T & 0 \end{bmatrix} \mid X \in Ut(n), t \in R^n\right\} \quad (6)$$

According to the matrix exponentials, triangular Matrix definitions and their properties, it is shown that S(n + 1) is the Lie algebra of the matrix group $S^+(n + 1)$. This is states that exp is a diffeomorphism from $S^+(n + 1)$ to its Lie algebra S(n + 1) and is proved in the following theorem-1 [22].

Theorem 1. The function exp : S(n + 1) → $S^+(n + 1)$, $S_{t, X} \rightarrow \exp(S_{t, X})$ is a smooth bijection and its inverse is smooth as well.

Proof: It is simple to prove that any $S_{t, X}$ in S(n +1) is uniquely mapped through the exponential function to $S^+(n + 1)$; conversely, any $S\mu, Z \in S^+(n + 1)$ has positive eigenvalues and thus log(Sμ, Z) uniquely exists in S(n+1) [22]. So exp : S(n + 1) → $S^+(n + 1)$ is one to one and onto, while the smoothness of exp and that of its inverse are guaranteed by [Theorem 1]. Log-Euclidean framework is extended to $S^+(n + 1)$ by the following theorem-2.

Theorem 2 (Log-Euclidean on $S^+(n + 1)$)**.** This theorem define

$$\otimes : S^+(n + 1) \times S^+(n + 1) \rightarrow S^+(n + 1),$$

$$S1 \otimes S2 = \exp(\log(S1) + \log(S2)), \text{ and}$$

$$\odot : R \times S^+(n + 1) \rightarrow S^+(n + 1),$$

$$\lambda \odot S = \exp(\lambda \log(S)) = S^{\lambda}.$$

Under operation ⊗, $S^+(n + 1)$ is a commutative Lie group,

log : $S^+(n + 1)$ → S(n + 1), S^+→ log(S) is a Lie group isomorphism. In addition, under operators ⊗ and ⊙, $S^+(n + 1)$ is a linear space.

Proof of this theorem is straightforward and is therefore omitted [note that S(n + 1) is a Lie group under *matrix addition*]. By far $S^+(n + 1)$ is equipped with a novel Lie group structure. The equivalence between $S^+(n+1)$ and S(n+1) is established by isomorphism , which indicates that the operations on $S^+(n+1)$ can be transformed, via matrix logarithm, to the linear space S(n + 1) while respecting the algebraic and topological structure of $S^+(n + 1)$. More general conclusion can be defined as follows. Let n × n be a real invertible matrices in a set IN(n) with nonnegative eigenvalues. Since any S ∈ IN(n) has a unique real logarithm, and

exp : log(IN(n)) → IN(n) is a diffeomorphsim, where log(IN(n)) denotes the image of IN(n) under logarithm, This can establish the Log-Euclidean on IN(n).

The complete embedding process is illustrated as follows:

$$\mathcal{N}(\mu, \Sigma) \overset{\Phi^{-1}}{\Longrightarrow} S_{\mu, L^{-T}} \overset{log}{\Rightarrow} log(S_{\mu, L^{-T}}) \quad (7)$$

In Eq -7 , $\Sigma = L^{-T}L^{-1}$ and $L^{-T} = PDUT(n)$. Recall that L is the Cholesky factor of $\sum^{-1}$.

Indirect Embedding

Indirect Embedding method consists of three consecutive functions.Let us consider the left coset of SO(n + 1) in $GL^+(n+1)$, ${}_PSO = \{PO|O \in SO(n + 1)\}$,

where P ∈ $Sym^+(n + 1)$ is an (n + 1) × (n + 1) SPD matrix. As |PO| = |P||O| = |P| > 0, where | · | denotes the matrix determinant, ${}_PSO$ is a subset of $GL^+(n + 1)$. Conversely, any matrix G ∈ $GL^+(n + 1)$ has a unique left polar decomposition G = PR, where P ∈ $Sym^+(n+1)$ and R ∈ SO(n + 1), and thus G belongs to one and only one coset ${}_PSO$. Hence, the set of cosets $\{{}_PSO, P \in Sym^+(n + 1)\}$ partitions $GL^+(n + 1)$ and this denote the set by the quotient $GL^+(n + 1)/SO(n + 1)$ which is wellknown to be a Lie group. Note that $S^+(n + 1)$ is a Lie subgroup of $GL^+(n + 1)$, and there exists an injective function for which

$$\pi : S^+(n + 1) \rightarrow GL^+(n + 1)/SO(n + 1), \quad (8)$$

π(S) $=_P$ SO, where S = PR is the left polar decomposition of S. Next, we map, through the bijective function

$$\gamma : GL^+(n + 1)/SO(n + 1) \rightarrow Sym^+(n + 1), \quad (9)$$

$\gamma({}_PSO)$ = P the coset ${}_PSO$ to the space of SPD matrices $Sym^+(n + 1)$. Below we show that γ is a Lie group isomorphism by defining an operation

$$* : GL^+(n + 1)/SO(n + 1) \times GL^+(n + 1)/SO(n + 1) \rightarrow GL^+(n + 1)/SO(n + 1)$$

$${}_PSO *_Q SO =_P \odot_Q SO, \quad (10)$$

where P ⊙ Q = exp(log(P) + log(Q)) is the logarithmic multiplication defined on $Sym^+(n + 1)$ [14]. The function γ is a Lie group homomorphism since $\gamma({}_PSO *_Q SO) = \gamma({}_P\odot_Q SO) = P \odot Q = \gamma({}_PSO) \odot \gamma({}_QSO)$. The smoothness of γ and that of its inverse are obvious and thus it is a Lie group isomorphism, which guarantees that $GL^+(n + 1)/SO(n + 1)$ be equivalent to $Sym^+(n + 1)$.

The third function aims to map the SPD matrices into a

linear space. To this end, we adopt the Log-Euclidean framework proposed by Arsigny et al. [23]. Their idea is to transform, via matrix logarithm, the Riemannian operations on $Sym^+(n+1)$ to the Euclidean ones in the vector space.

Theorem 3 (Log-Euclidean on $Sym^+(n+1)$)**.** Under operation $\otimes$, $Sym^+(n+1)$ is a commutative Lie group, and

$$\log : Sym^+(n+1) \rightarrow Sym(n+1),\ P \rightarrow \log(P)$$

is a Lie group isomorphism[23].

In addition, under $\otimes$ and $\odot$, $Sym^+(n+1)$ is a linear space. Thus far, to summarize the complete embedding process of IE-LogE as follows:

$$\mathcal{N}(\mu,\Sigma) \overset{\Phi^{-1}}{\Longrightarrow} S_{\mu,L^{-T}} \overset{\pi}{\Rightarrow} p^{SO} \overset{\gamma}{\Rightarrow} P \overset{\log}{\Rightarrow} \log(P) \qquad (11)$$

Here $\sum = L^{-T}L^{-1}$ and L is the Cholesky factor of $\sum^{-1}$; $S_{\mu,L-T}$ has left polar decomposition $S_{\mu,L-T} = PR$.

Properties of P **in (11)** This matrix has two properties.

1. It is the square root matrix of $S_{\mu,L-T} S^T_{\mu,L^{-T}}$ i.e

$$P = \begin{bmatrix} \Sigma + \mu\pi^T & \mu \\ \mu^T & 1 \end{bmatrix}^{1/2}$$

The eigenvalues of P are identical to the singular values of $S_{\mu,\ L-T}$. In addition, their $\ell 2$-norm condition numbers are identical.

2) The matrix R that accompanies P is the closest possible orthogonal matrix to $S_{\mu,L^{-T}}$. That is

$$R = \text{argmin}_{O \in O(n+1)} \| S_{\mu,L^{-T}} - O \|_F, \qquad (12)$$

where $\| \cdot \|_F$ denotes the Frobenius norm and O(n + 1) is the orthogonal group of dimension n + 1.

Embedding by right coset In previous developments it has accomplished the embedding of Gaussians based on the left coset of SO(n + 1). In a very similar manner, we can consider the right coset $SO_P = \{OP | O \in SO(n+1)\}$ to obtain the second embedding scheme. The space of cosets $\{SO_P, P \in Sym+(n+1)\}$ partitions $GL^+(n+1)$, and we denote this space by $GL^+(n+1)\backslash SO(n+1)$. Note that any invertible matrix has a unique *right* polar decomposition. We map a matrix $S \in S^+(n+1)$ to an SPD matrix through the following two functions:

$$\tilde{\pi} : S^+(n+1) \rightarrow GL^+(n+1)\backslash SO(n+1),\ \tilde{\pi}(S) = SO_{P'},$$

$$\tilde{\gamma} : GL^+(n+1)/SO(n+1) \rightarrow Sym^+(n+1),\ \tilde{\gamma}(SO_{P'}) = P'.$$

Here $S = R'P', R' \in SO(n+1), P' \in Sym+(n+1)$ is the right polar decomposition of S. In this case, the embedding matrix P′ is the square root of $S^T_{\mu,L^{-T}}\mu$, $S_{\mu,L-T}$, i.e.,

$$P' = \begin{bmatrix} L^{-1}L^{-T} & L^{-1}\mu \\ \mu^T L^{-T} & \mu^T\mu + 1 \end{bmatrix}^{\frac{1}{2}}. \qquad (13)$$

Based on the left coset and right coset, we obtain the SPD matrices (11) and (13), respectively. However, this embedding mechanism is different from theirs; most importantly, we further map SPD matrices into the linear space to handle Gaussians with Euclidean operations.

LOG-EUCLIDEAN MULTIVARIATE GAUSSIAN DESCRIPTORS

For a given image M, we first extract features through densely sampling in a regular grid or using an interest point detector. Let $X = \{x_1 : : : x_N\}$ be a set of local features in M which are extracted using SIFT algorithm[9]. These features are again described with Log Euclidean Multivariate Gaussian distribution [16], [18] supposing that they are normally distributed. Image retrieval framework is shown in the Fig- 2.

The Multivariate Gaussian distribution [4] of a set of d-dimensional vectors X is calculated by Eq-4 i.e.

$$f(X|\xi,K) = (2\pi)^{-d/2}(detK)^{1/2}e^{-(x-\xi)^T K(x-\xi)/2} \qquad (14)$$

where j is the determinant, ξ is the mean vector and K is the covariance matrix with space of real symmetric positive semi-definite matrices [6] defined as follows:

$$\xi = \frac{1}{N}\sum_{i=1}^{N} x_i\,, \qquad K = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \xi)(x_i - \xi)^T$$

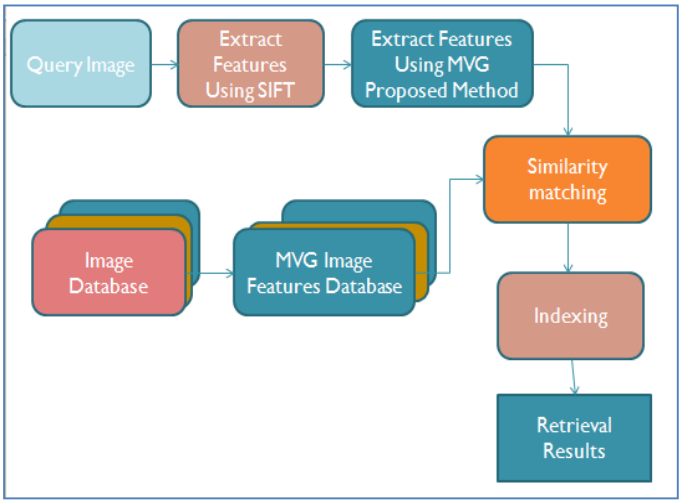

Figure 2: Image retrieval Frame Work Using MG

The covariance matrix encodes information about the variance of the features and their correlation. Although it is very informative, it does not lie in a vector space since the covariance space is not closed under multiplication with a negative scalar. In fact, it lies in a Riemannian manifold. Most of the common machine learning algorithms assume that the data points form a vector space, therefore a suitable transformation is required prior to their use. Since the

covariance matrix is symmetric positive definite we adopt the Log-Euclidean metric shown in theorems 1, 2,3. The basic idea of the Log-Euclidean metric is to construct an equivalent relationship between the Riemannian manifold[15] and the vector space of the symmetric matrix. In [10] an approach to map from Riemannian manifolds to Euclidean spaces is described. The first step is the projection of the covariance matrices on an Euclidean space tangent to the Riemannian manifold, on a specific tangency matrix K. The second step is the extraction of the orthogonormal coordinates of the projected vector. In the following, matrices (points in the Riemannian manifold)[13], [15] will be denoted by bold uppercase letters, while vectors (points in the Euclidean space) by bold lowercase ones. More formally, the projected vector of a covariance matrix K is given by:

$$t_K = log_p(K) = P^{\frac{1}{2}}\log(P^{-\frac{1}{2}}KP^{\frac{1}{2}})P^{\frac{1}{2}} \quad (15)$$

Where log is the matrix logarithm operator and log(P) is the manifold specific logarithm operator, dependent on the point P to which the projection hyper plane is tangent. The matrix logarithm operators of a matrix K can be computed by eigenvalue decomposition ($K = UDU^T$); it is given by:

$$\log(K) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n}(K-I)^n = Ulog(D)U^T \quad (16)$$

The orthogonormal coordinates of the projected vector t_K in the tangent space at point P are then given by the vector operator:

$$\mathrm{vec_p}(\mathrm{t_K}) = \mathrm{vec_I}\left(\mathrm{P}^{-\frac{1}{2}}\mathrm{t_K}\mathrm{P}^{-\frac{1}{2}}\right) \quad (17)$$

where I is the identity matrix, while the vector operator on the tanget space at identity of a symmetric matrix Y is defined as:

$$\mathrm{vec_I(Y)} = \left[y_{1,1}\ \sqrt{2}\,y_{1,2}\ \ \sqrt{2}y_{1,3}\ y_{2,2}\ \sqrt{2}\,y_{2,3}\ \ldots\ldots y_{d,d}\right] \quad (18)$$

Substituting t_K from Eq. 16 in Eq. 18, the projection of C on the hyper plane tangent to P becomes

$$\mathrm{K} = \mathrm{vec_I}\left(\log\left(\mathrm{P}^{-\frac{1}{2}}\mathrm{K}\mathrm{P}^{-\frac{1}{2}}\right)\right) \quad (19)$$

Thus, after selecting an appropriate projection origin, every covariance matrix is projected to an Euclidean space. Since K is a symmetric matrix of size d X d a $(d^2 + d)/2$-dimensional feature vector is obtained.

The projection point P is arbitrary and, even if it could influence the performance (distortion) of the projection, from a computational point of view, the best choice is the identity matrix, which simply translates the mapping into a standard matrix logarithm. In short, our method is to extract local descriptors from an image and then collect them in a spatial pyramid; each sub-region is described by a multivariate Gaussian distribution. The covariance matrix is projected on a Euclidean space and concatenated to the mean vector to obtain the final descriptor. We empirically observe that most of the values in the concatenated descriptor are low, while few are high. In order to distribute the values more evenly, we adopt the power normalization method. i.e. to apply to each dimension the function f(x) = sign(x) jxj with = 0:5: Eventually, the concatenated descriptors are fed to a linear classifier.

RESULTS ON BENCHMARK DATASETS

We carry out experiments on popular benchmark datasets under the MVG framework but focus on descriptor comparison. Our experiments involve object recognition on Caltech-256 [19], Caltech-101, scene recognition on Scene-15 [20], Coral (Wang) Database.

Caltech-256 [19] has about 30K images distributed in 256 categories, containing diverse object sizes, poses, and lighting conditions. We present the average recognition accuracy over five trials in Fig.3. We compare with the methods in [9], [24], all of which involve mapping or transforming of the local features (e.g., SIFT) and achieve improved performance. With Log Euclidean Multivariate Gaussian Descriptors outperforms all the forementioned competing methods, even with PCA-SIFT.

Caltech-101 has about 9008 images distributed about 101 categories and snapshot this database is shown in Fig.4

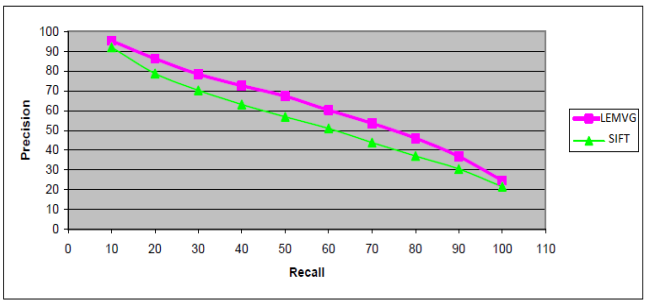

Figure 3: Average Precision of SIFT method and Multivariate Gaussian Method at different level of recall

Scene-15 [20] consists of 4,485 images, and the number of images per category varies from 200 to 400. The snapshot of Scene-15 is shown in Fig-5. Following [20], we randomly choose 100 training images per class, while the remaining ones are reserved for testing. The average accuracy over five trials is reported.

Figure 4: Snapshot of Caltech 101 Image Database

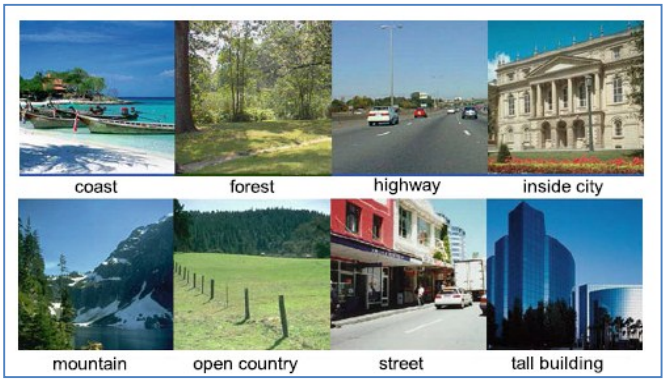

Figure 5: Snapshot of Scene -15 Database

Out of 256 categories from Caltech 256 image database 10 images from each category are used as queries. For each query, average precision(AP) of the retrieval at each level of the recall is obtained. The retrieval accuracy average precision of SIFT method at each level of recall is presented in table 1. The retrieval accuracy of the proposed method LEMVG is presented in table 2. With Log Euclidean Multivariate Gaussian we further achieve average precision improvement of 8.96% on average.

Table 1

Object Recognition Accuracy(AP, %) of SIFT Method

Category	*10%*	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*	*80%*	*90%*	*100%*	
airplanes–251	80	80	66.71	66.69	66	57.78	56.19	25	13.9	12.43	
leopards–129	80	75.5	80	72.91	57.68	41.64	41.43	31.43	30.3	26.54	
sunflower–204	80	73.45	70.91	69.75	53.68	41.64	31.43	30.22	23.45	21.87	
motorbikes–145	80	75.5	72.34	70.65	55.54	34.65	30.43	28.53	27	18.81	
tombstone-222	70	65.56	71.43	70.65	60.28	54.87	51.61	30.33	22.43	20.21	
waterfall-241	80	75.5	72.32	71.63	62.93	55.92	50.61	34.64	18.67	16.59	
binoculars-012	75	70.54	63.82	60.67	58.95	50.56	48.25	40.62	32.37	16.43	
greyhound-254	75	78.32	71.43	69.75	55.54	54.87	41.43	30.62	18.67	7.54	
traffic-light-226	55	37.65	36.76	22.59	20.28	12.18	10.51	7.39	3.34	0.36	
car-side-252	80	67.4	37.23	32.93	30.31	26.32	23.52	20.05	18.42	14.32	
	75.5	69.94	58.572	55.53	47.088	38.41	34.19	23.58	17.01	12.08	43.2154

Table 2

Object Recognition Accuracy(AP, %) of LEMVG Method

Category	*10%*	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*	*80%*	*90%*	*100%*	
airplanes–251	85	85	70.54	69.59	68.57	62.78	60.42	25.55	17.73	12.48	
leopards–129	85	85	78.58	72.48	61.29	47.84	46.43	36.54	33.45	27.58	
sunflower–204	80	80	77.49	73.45	57.56	45.35	36.43	32.66	25.46	53.32	
motorbikes–145	85	83.32	75.5	74.48	39.67	39.65	35.43	32.23	31.67	25.53	
tombstone-222	85	79.65	70.3	69.62	63.52	57.45	55.53	41.77	35.34	23.46	
waterfall-241	85	85	77.32	76.63	67.93	60.92	55.61	39.63	23.67	21.59	
binoculars-012	80	75.34	68.82	65.67	63.95	55.56	53.25	45.36	37.37	21.43	
greyhound-254	85	83.32	76.43	74.75	60.54	59.87	46.43	35.32	23.67	12.54	
traffic-light-226	51.67	53.47	35.56	31.56	23.65	23.46	23.45	11.3	9.56	7.45	
car-side-252	85	52.34	55.45	50.53	45.56	43.56	33.45	25.35	21.41	18.78	
	80.667	76.24	68.599	65.88	55.224	49.64	44.64	32.56	25.93	22.42	52.1816

CONCLUSION

This paper presented a function-valued descriptor called Log Euclidean Multivariate Descriptors to characterize local, high-order statistics by extracting Gaussian distributions from a local neighborhood.

We developed Log-Euclidean methods to handle Gaussians with Euclidean operations instead of Riemannian ones. Unlike popular histogram-based descriptors, which, based on feature space quantization, collect zero-order (occurrence) information, the proposed Log Euclidean Multivariate Descriptors is continuous and models higher-order statistics. It can naturally leverage multiple cues or other descriptors (e.g, SIFT) as raw features. We showed that the space of Gaussians can be equipped with a Lie group structure, and that it is equivalent to a subgroup of the upper triangular matrix group. These conclusions, not presented in previous literature as far as we know, provide new insights into the algebraic and geometrical structure of Gaussians. Compared to histogram-based descriptors (e.g. SIFT), our Log Euclidean Multivariate Descriptors are computationally more demanding.

REFERENCES

[1] T.Venugopal, B.Ramesh Naik, V.Kamakshi Prasad, “Image retrieval using adapted Fourier Descriptors” in Int. J. Signal and Imaging Systems Engineering, Vol. 3, No. 3, pp. 188-194, 2010.

[2] Persoon, E. and Fu, K.S. ‘Shape discrimination using Fourier descriptors’, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 3, March, pp.170–179. 1997

[3] Hailong Shi, Hao Zhang, Gang Li,Xiqin Wang , “ Stable embedding of Grassmann Manifold via Gaussian Random Matrices”, IEEE Transaction on Information Theory, Vol. 61, No. 5 pp. 2924- 2924, May 2015.

[4] Miquel Calvo., Josep M. Oller “A distance between

multivariate normal distributions based in an embedding into the siegel group", Journal of Multivariate Analysis Volume 35, Issue 2, pp. 223-242 November 1990.

[5] Peihua Li, Qilong Wang, Hui Zeng, Lei Zhang "Local Log Euclidean Multivariate Gaussian Descriptors and Its application to Image Classification", IEEE Transaction on PAMI – Volume: 39, Issue: 4, pp.803-817, 2017.

[6] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite atrices," SIAM J. Matrix Anal. Appl, 2006.

[7] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: Theory and practice," Int. J. Comput. Vis., vol. 105, no. 3, pp. 222–245, 2013.

[8] D. L. Bihan, J. Mangin, C. Poupon, C. Clark, S. Pappata, and N. Molko, "Diffusion tensor imaging: Concepts and applications," J Magn. Reson. Imaging, vol. 66, pp. 534–546, 2001.

[9] D. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vision, vol. 60, pp. 91–110, 2004.

[10] G. Serra, C. Grana, M. Manfredi, and R. Cucchiara, "Modeling local descriptors with multivariate Gaussians for object and scene recognition," in Proc. ACMInt. Conf.Multimedia, 2013, pp. 709–712.

[11] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2004, pp. II–506.

[12] K. van de Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1582–1596, Sept 2010.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. Int. Conf. Comp. Vis. Patt. Recog., 2005, pp. 886–893.

[14] T.Venugopal, B.Ramesh Naik, V.Kamakshi Prasad, "Adapted Zernike moments Based Image Retrieval" in The Journal of Computing, (TJC) Vol. 1 Issue. 5 Sep-2010.

[15] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," Int. J. Comput. Vision, pp. 41–66, 2006.

[16] L. Gong, T. Wang, and F. Liu, "Shape of Gaussians as feature descriptors," in Proc. Int. Conf. Comp. Vis. Patt. Recog., 2009, pp.2366–2371.

[17] Bergstra, J. and Bengio, Y. "Random search for hyper-parameter optimization". Journal of Machine Learning Research, 13:281–305, 2012.

[18] UIUC, Lecture 21. The Multivariate Normal Distribution, 21.5: "Finding the Density".

[19] G. Griffin, A. Holub, and P. Perona, "The Caltech-256," California Institute of Technology, Tech. Rep., 2007.

[20] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2006, pp. 2169–2178.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients forhuman detection," in Proc. Int. Conf. Comp. Vis. Patt. Recog., 2005, pp. 886–893

[22] Hall, B. C. Lie groups, Lie algebras, and Representations: An Elementary Entroduction. Graduate Texts in Mathematics. **222** (2nd ed.). Springer. doi:10.1007/978-3-319-13467-3. ISBN 978-3319134666 ISSN 0072-5285. 2015

[23] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: Theory and practice," Int. J. Comput. Vis., vol. 105, no. 3, pp. 222–245, 2013.

[24] T. Kobayashi, "Dirichlet-based histogram feature transform for image classification," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 3278–3285.

[25] B.Ramesh Naik, T.Venugopal, "Efficient object Recognition using Descriminative Weight Learning", Int. Journal of Computer Engineering and Technology, Vol 8, issue 4, pp. 28-35 Jun 2017