Flip-Invariant Motion Representation

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

Abstract

In action recognition, local motion descriptors contribute to effectively representing video sequences where target actions appear in localized spatio-temporal regions. For robust recognition, those fundamental descriptors are required to be invariant against horizontal (mirror) flipping in video frames which frequently occurs due to changes of camera viewpoints and action directions, deteriorating classification performance. In this paper, we propose methods to render flip invariance to the local motion descriptors by two approaches. One method leverages local motion flows to ensure the invariance on input patches where the descriptors are computed. The other derives a invariant form theoretically from the flipping transformation applied to hand-crafted descriptors. The method is also extended so as to deal with ConvNet descriptors through learning the invariant form based on data. The experimental results on human action classification show that the proposed methods favorably improve performance both of the hand-crafted and the ConvNet descriptors.

1. Introduction

There is an increasing amount of multimedia data containing videos through security cameras in the real world and web sites (such as YouTube) on the Internet. Thereby, it creates an urgent demand for automatic action recognition in computer vision communities. The action recognition has been tackled over the last two decades [40, 28, 11]. The difficulty of the action recognition is first in extracting effective motion features. An input video is formulated in a spatio-temporal volume while the images are defined in a two-dimensional space domain. Such higher dimensionality of input data makes it harder to design motion features.

Along with the advances of image classification, the motion descriptors which extract motion characteristics are developed in the framework of bag-of-features over spatio-temporal interest points and/or dense trajectories, exhibiting successful performance in realistic videos [20, 6, 35, 36]. On the other hand, deep convolutional neural network (ConvNet) methods have been applied to various image recognition tasks with great success, and it is now being extended to motion recognition fields together with the large-scale video dataset [17, 16, 32]. The ConvNet methods can construct spatio-temporal features to effectively describe the motion patterns via end-to-end learning [32]. These two approaches are comparable from the viewpoint of classification accuracy, being different from the image classification where the ConvNet significantly outperforms the hand-crafted methods, which attracts enthusiastic research effort to improve ConvNet methods in terms of architecture [29], training scheme [38] and local descriptors [37].

An effective descriptor is required to be robust or invariant against variations of input signal which are irrelevant to recognition, and in action classification, it is necessary to acquire the invariance to *horizontal (mirror) flipping* in video sequences. Both video cameras and actors (humans) are standing upright due to gravity on the Earth and thus the rotation around the optical axis of the camera rarely occurs. On the other hand, the horizontal flipping is frequently observed such as due to changes of the camera viewpoint (from the front or the back) and/or the action direction (leftward or rightward); through horizontal flipping, the video frames capturing leftward walking correspond to those of rightward walking captured from the opposite side, and vice versa. Robust recognition can be built on the motion descriptors that characterize motions while being invariant to the horizontal flipping. Though some invariant descriptors are proposed in the image domain (see a brief review in Sec. 1.1), such flip invariance has not been well addressed in the action recognition literature.

In this paper, we propose methods to improve performance of action classification by taking into account the invariance against the horizontal flipping. We focus on local motion descriptors, including ConvNet ones [37], which work well in the trajectory-based BoF framework [36] for action classification. It is feasible to consider the invariance on a local region of spatio-temporal volume, while global invariance is hard to be treated in a holistic video descrip-

tor containing backgrounds and roughly global locations of the targets. In the proposed methods, the flip invariance is achieved from two perspectives of patch level and descriptor level. The first one considers the invariance on input patches before extracting descriptors. The input patches can be invariant to the flipping by means of local motion flows so that the descriptors extracted from those patches are invariant. In the second approach, we propose an invariant form of the local motion descriptors themselves. The flipping can be explicitly represented by linear transformation for the hand-crafted descriptors, such as HOF [20] and MBH [6], and thereby the invariant form is analytically derived from the explicit transformation. In contrast, the ConvNet local descriptors [37] of deep non-linearity do not provide such explicit representation for the flipping. Thus, the method is extended so as to efficiently learn the invariant form based on data for that kind of descriptors whose transformation is not known a priori. The proposed method transforms the descriptors into the effective form of invariance against the flipping with a low computational cost. In addition to those methods toward flip invariance, we also present a simple yet efficient hand-crafted motion descriptor which is not based on histogram features, and show that the proposed flip-invariant methods work on various types of motion descriptors including it.

1.1. Related works

For action recognition, view-invariance has been addressed via 3D (XYZ) representation [41, 24, 39] beyond (2D) image sequences and it is also discussed from the biological viewpoint [14]. Flip invariance, however, has not been discussed so well for the video descriptors. On the other hand, in the image domain, some methods are proposed to embed flip invariance into image descriptors, *e.g.*, SIFT [21], mainly for robust image matching [45, 42, 9, 43, 10, 22]. Those methods can be categorized into two groups from the viewpoint of whether the invariance is achieved before or after computing descriptors.

One approach is to make an input patch flip-invariant by normalizing it; that is, the patch is 'flipped' when the flipping transformation is observed. The image descriptor extracted on such an invariant patch is thus invariant against the flipping. The key issue is how to determine whether a patch should be flipped or not, and in the image domain, a criterion for the determination is based on image gradients. Max-SIFT [42] utilizes one component of SIFT features, and in MIFT [10] and FIND [9], some of orientation bins related to horizontal direction contribute to determining the path direction. FIND [9] further arranges spatial bins in a more efficient manner to produce effective feature ordering suitable for the flipping. F-SIFT [45] is proposed by leveraging curl [26] operator to enforce that the gradient flow in a patch follows the pre-defined direction, and RIDE [43] simply employs sum of the horizontal image gradients.

While most methods toward flip invariance are formulated by the above approach, there is the other way to achieve the invariance after computing the descriptors. MI-SIFT [22] directly transforms the SIFT descriptors by applying component-wise operation, such as averaging, to two descriptors extracted on the original and flipped patches; the one on the flipped patch can be computed in an efficient manner by swapping original SIFT components.

Those methods in the image domain are intended for keypoint matching by means of local descriptors and validated in the matching task except for [43]. In this study, we propose methods to achieve the flip invariance of motion descriptors for *action* classification.

The above-mentioned approaches are computationally efficient only for hand-crafted descriptors, such as SIFT, whose flipped counterparts can be analytically computed. It does not hold for the more complicated descriptor, such as by ConvNet [37], whose flipped one is obtained only by re-computation on the flipped patches, which doubles the computational cost; even though the first approach works before computing descriptors, from a practical viewpoint, it is more efficient to apply a holistic method, such as convolution, to videos *twice* for obtaining the flipped descriptors than to apply a patch-wise computation (with flipping). Therefore, we theoretically derive an invariance method and extend it to efficiently deal with the ConvNet descriptors.

In the other way, flip invariance is naively treated by augmenting datasets through actually flipping images/videos [3, 2], though doubling the computational cost and memory consumption both in training and test phases.

2. Local motion descriptor

We first show the local motion descriptors that are employed in this study as ingredients to represent a video sequence in the BoF framework [36].

Optical flow is fundamental to characterize motions in a video sequence, and thus we convert the video frames into flow field images composed of two components, horizontal and vertical flow velocities denoted by $f_x(\boldsymbol{p})$ and $f_y(\boldsymbol{p})$ at pixel $\boldsymbol{p}$, respectively. As in [35, 36, 15, 44], we employ HOF [20] and MBH [6] to extract features from the flows and their derivatives, respectively, in addition to HOG [5] for describing appearance. From the viewpoint of extracting features on the multi-channeled images, the motion descriptors resemble those of *colors* which are also extracted from colored images of three (RGB) channels. Table 1 summarizes the relationships between the descriptors proposed in those two domains and makes us realize that in the motion domain there is a missing counterpart to LCS [4] which is based on local statistics of raw color signals. Thus, in this study we present a local motion descriptor based on *local motion statistics* (LMS) in accordance with the LCS

Table 1. Relationships between color and motion descriptors.

	histogram on raw channels	histogram on derivatives	local statistics
Color	Color-histogram [31]	RGB-SIFT [33]	LCS [4]
Motion	HOF [20]	MBH [6]	(*LMS*)

color descriptor. LMS is simply built on the mean and standard deviation of the flow components $\{f_x, f_y\}$ to form four-dimensional features $\{\mu_x, \mu_y, \sigma_x, \sigma_y\}$ in each grid on a spatio-temporal patch; we divide the spatio-temporal patch extracted along the trajectory [35, 36] into $2_{(X)}\times 2_{(Y)}\times 3_{(T)}$ for HOG, HOF and MBH, and $3_{(X)}\times 3_{(Y)}\times 3_{(T)}$ for LMS so as to produce descriptors of the similar dimensionality (Fig. 1). Though both LMS and HOF are based on the raw flow velocities, LMS describes them by using the statistics without orientation coding, and therefore it has the advantage of compensating the other histogram-based descriptors as well as fast computation.

In addition to the above *hand-crafted* descriptors, we employ the *learning*-based descriptors [37] which leverages spatial and temporal ConvNets [29] to extract local features; for details of the ConvNets and feature extraction process, refer to the paper [37].

2.1. Flipped local descriptor

Next, we explain how the hand-crafted local descriptors are transformed by flipping an input video. The horizontal flipping in videos changes flow as well as appearance, which is slightly complicated compared to flipping images. The flipping affects the processes of computing descriptors in the following three points. First, spatio-temporal grids in a patch are horizontally swapped. Second, the derivative along the horizontal axis is negated with *sign* inversion; $\partial_x \hat{\mathcal{I}}(\hat{\boldsymbol{p}}) = -\partial_x \mathcal{I}(\boldsymbol{p})$ where $\hat{\mathcal{I}}$ and $\hat{\boldsymbol{p}}$ indicate the flipped image frame and position of $\mathcal{I}$ and $\boldsymbol{p}$, respectively. Third, the flow orientations are horizontally flipped, inverting the sign of the horizontal flow velocity, $\hat{f}_x(\hat{\boldsymbol{p}}) = -f_x(\boldsymbol{p})$, while keeping the vertical flow velocity the same, $\hat{f}_y(\hat{\boldsymbol{p}}) = f_y(\boldsymbol{p})$. In addition, the derivatives of flows are also transformed by

$$\begin{bmatrix} \partial_x \hat{f}_x(\hat{\boldsymbol{p}}) & \partial_x \hat{f}_y(\hat{\boldsymbol{p}}) \\ \partial_y \hat{f}_x(\hat{\boldsymbol{p}}) & \partial_y \hat{f}_y(\hat{\boldsymbol{p}}) \end{bmatrix} = \begin{bmatrix} \partial_x f_x(\boldsymbol{p}) & -\partial_x f_y(\boldsymbol{p}) \\ -\partial_y f_x(\boldsymbol{p}) & \partial_y f_y(\boldsymbol{p}) \end{bmatrix}, \quad (1)$$

where for example $\partial_y f_x(\boldsymbol{p})$ indicates the vertical derivative (∂_y) of the horizontal flow component f_x at the position $\boldsymbol{p}$.

On the basis of the above analysis, we can explicitly derive the flip transformation for the hand-crafted descriptors as follows (Fig. 1). For the descriptors based on orientation histograms, HOG, HOF and MBH, the flip transformation is defined as swapping feature components according to the flipping of grid positions and the orientation bins. As to LMS, the transformation is composed of swapping the components of μ_x, μ_y, σ_x and σ_y according to the grid flipping with inverting the sign of μ_x which is the only statistics affected by horizontal flipping.

On the other hand, it is difficult to explicitly derive the transformation of the ConvNet descriptors [37] since the input patch is processed in a highly non-linear manner through the ConvNet. In that case, it is inevitable to re-compute the ConvNet descriptors on the flipped input videos for obtaining the flipped descriptors.

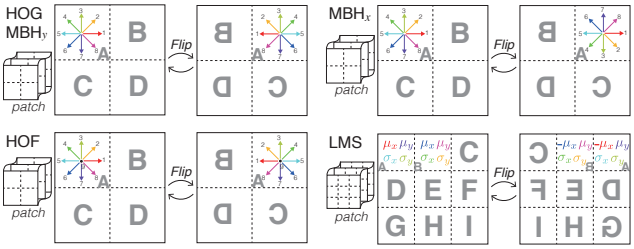


Figure 1. Transformation of the hand-crafted descriptors by flipping an input video. The spatio-temporal grids on a patch are indicated by $A \sim I$. We employ 8 orientation bins for histogram-based descriptors and additionally 1 bin for null flow in HOF. The colors show the correspondence of feature components. Note that MBH$_x$ descriptor components are *vertically* swapped due to the relationship (1) and the feature component μ_x in LMS is negated.

Table 2. Summary of the proposed methods.

Invariance level	Feature type	Transformation procedure	Comp. overhead
Desc.-level	hand-craft	Eq.5 with sparse $\boldsymbol{w}_i$ (Eq.8, 10)	$O(m)$
Desc.-level	ConvNet	Eq.5 with dense $\boldsymbol{w}_i$ learned by Alg.1	$O(m^2)$
Patch-level	hand-craft	Flip by Fig. 1 based on $\mathtt{sign}\{\sum f_x(\boldsymbol{p})\}$	$O(m)$
Patch-level	ConvNet	Recompute feature on a flipped video and use it based on $\mathtt{sign}\{\sum f_x(\boldsymbol{p})\}$	Recomp.

3. Flip-invariant representation

We propose methods to render invariance against horizontal flipping for the local motion descriptors. The methods are formulated in two ways of patch level and descriptor level, considering the invariance before or after computing descriptors. As a result, there are 4 types of methods: {patch-level invariance (Sec. 3.1), descriptor-level invariance (Sec. 3.2)} $\times$ {hand-crafted, ConvNet}-descriptors, as summarized in Table 2.

3.1. Patch-level invariance

Before descriptor computation, it is possible to achieve invariance by making the input *patch* invariant to flipping. The key issue is how to determine the leftward/rightward orientation of the patch; once the patch orientation is determined, we can flip the input patch such that its orientation is aligned, *e.g.*, to be rightward. The previous works [45, 42, 9, 43, 10] on the image domain estimate the orientation based on image gradients over the patch region $\mathcal{D}$, *e.g.*, mean of the horizontal gradient component, $\mathtt{sign}\{\sum_{\boldsymbol{p}\in\mathcal{D}} \partial_x \mathcal{I}(\boldsymbol{p})\}$ [43], to provide flip invariance in terms of *shape*. We propose a method to lever-

age motion flows for estimating the horizontal direction of the patch. The method directly utilizes the horizontal flow velocity in the patch and thereby the patch direction is determined by its sign, $\texttt{sign}\{\sum_{\boldsymbol{p}\in\mathcal{D}} f_x(\boldsymbol{p})\}$; the proposed method achieves flip invariance regarding *motion flow* in contrast to the previous ones. We call this method by 'Flip-by-Flow'. As an alternative, considering that the spatio-temporal patch is extracted along the trajectory [35], the dominant direction is also estimated by fitting a line into a sequence of x-position of the trajectory; $\texttt{sign}\{\arg_\alpha \min_{\alpha,\beta} \sum_t (x_t - \alpha t - \beta)^2\}$ (line fitting) where x_t is the x-coordinate of trajectory at time t, and we call this 'Flip-by-Trajectory'.

Both methods based on gradients and flows achieve invariance against flipping in local descriptors. However, the flow-based patch direction estimation is advantageous in the following two points. First, it reduces false positives. The image gradient orientation $\texttt{sign}(\partial_x \mathcal{I})$ is vulnerable to changes of illumination and local contrast which occur independently of flipping. This causes false-positive decision on $\texttt{sign}\{\sum \partial_x \mathcal{I}(\boldsymbol{p})\}$. Second point is that the dominant flow is stably estimated on the local spatio-temporal patch. The motion flows are not significantly fluctuated but smoothly distributed in local patches especially due to human kinematics. In contrast, the image gradients are diversely distributed due to the object of complex shape, making the patch direction estimation unstable.

Although these methods work before computing descriptors in essence, it practically requires re-computation for the flipped descriptors in the case that an efficient holistic approach is applied to compute descriptors, such as ConvNet [37]; the patch-wise computation with flipping is even slower than the (doubled) holistic process. On the other hand, it is computationally efficient to compute the hand-crafted descriptors on the flipped patches since only swapping components (Fig. 1) is applied to the pre-computed descriptors with negligible computation cost.

3.2. Descriptor-level invariance

The horizontal flipping of our interest is simpler than the other variations such as sift and rotation. It is formulated as *1-bit* (on/off) transformation without continuous parameters, *e.g.*, shift displacement and the degree of rotation, as described in Sec. 2.1, and is mathematically represented by

$$\hat{\boldsymbol{d}} = \boldsymbol{T}^\top \boldsymbol{d}, \tag{2}$$

where $\boldsymbol{d} \in \mathbb{R}^m$ and $\hat{\boldsymbol{d}}$ indicate a descriptor and its flipped one, respectively, and $\boldsymbol{T} \in \mathbb{R}^{m\times m}$ is the transformation matrix subject to $\boldsymbol{T}\boldsymbol{T} = \boldsymbol{I}$ indicating $\boldsymbol{T}$ is an involutory matrix [12]. From the transformation (2), the invariant form can be derived as linear operation with the vector $\boldsymbol{w} \in \mathbb{R}^m$ which satisfies

$$\boldsymbol{w}^\top \hat{\boldsymbol{d}} = \boldsymbol{w}^\top \boldsymbol{T}^\top \boldsymbol{d} = \boldsymbol{w}^\top \boldsymbol{d}. \tag{3}$$

This should hold for any descriptors $\boldsymbol{d}$, and thus the condition is simplified into

$$\boldsymbol{T}\boldsymbol{w} = \lambda \boldsymbol{w},\ s.t.\ \lambda = 1, \tag{4}$$

which states that $\boldsymbol{w}$ is the eigenvector of the eigenvalue $\lambda = 1$; this is the special case of invariant subspace [1].

In general, however, the flipping transformation matrix $\boldsymbol{T}$ contains the other eigenvalue than $\lambda = 1$; theoretically speaking, the eigenvalues of $\lambda = \pm 1$ exist in $\boldsymbol{T}$. The eigenvectors associated with the negative eigenvalue ($\lambda = -1$) do not exhibit any invariance, but discarding them leads to deteriorating descriptive power of the descriptors since the matrix $\boldsymbol{T}$ of full rank is constructed by the whole set of the eigenvectors including them. Thus, we leverage all the eigenvectors $\{\boldsymbol{w}_i\}_{i=1}^m$ to construct the invariant form as

$$\mathrm{g}_i(\boldsymbol{d}) = \begin{cases} \boldsymbol{w}_i^\top \boldsymbol{d} & \lambda_i \geq 0 \\ |\boldsymbol{w}_i^\top \boldsymbol{d}| & \lambda_i < 0 \end{cases},\ i \in \{1, \cdots, m\}, \tag{5}$$

where $|\cdot|$ produces the absolute value. Obviously, the latter form for $\lambda_i = -1 < 0$ is invariant against the flipping by

$$\mathrm{g}_i(\boldsymbol{T}^\top \boldsymbol{d}) = |\boldsymbol{w}_i^\top \boldsymbol{T}^\top \boldsymbol{d}| = |-\boldsymbol{w}_i^\top \boldsymbol{d}| = |\boldsymbol{w}_i^\top \boldsymbol{d}| = \mathrm{g}_i(\boldsymbol{d}). \tag{6}$$

The proposed invariant form employing all the eigenvectors loses only *sign* information of the projection by the eigenvectors $\{\boldsymbol{w}_i\}_{i|\lambda_i<0}$. Note again that the flip transformation matrix $\boldsymbol{T}$ contains only eigenvalues of either $\lambda = \pm 1$ and thus the method realizes complete invariance, $\mathrm{g}(\hat{\boldsymbol{d}}) = \mathrm{g}(\boldsymbol{d})$. We show the specific invariant forms for the hand-crafted motion descriptors (Sec. 2) as follows.

As described in Sec. 2.1, the flip transformation for the hand-crafted descriptors of HOG, HOF, $\mathrm{MBH}_{x/y}$ and LMS excluding μ_x swaps a pair of feature components $\{d_i, d_j\}$ of flip-correspondence, which is represented by

$$\boldsymbol{T}_{(i,j)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{7}$$

whose eigenvectors and eigenvalues are

$$\boldsymbol{w}_1 = \frac{1}{\sqrt{2}}[1,1]^\top, \lambda_1 = 1, \quad \boldsymbol{w}_2 = \frac{1}{\sqrt{2}}[1,-1]^\top, \lambda_2 = -1. \tag{8}$$

The eigenvector $\boldsymbol{w}_1$ of $\lambda_1 = 1$ operates as averaging, like the previous method [22], while $\boldsymbol{w}_2$ of $\lambda_2 = -1$ extracts the difference between the feature components $|d_i - d_j|$.

As to LMS, the components related to μ_x are not only swapped but also negated by the transformation matrix of

$$\tilde{\boldsymbol{T}}_{(i,j)} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, \tag{9}$$

whose eigenvectors and eigenvalues are

$$\tilde{\boldsymbol{w}}_1 = \frac{1}{\sqrt{2}}[1,1]^\top, \tilde{\lambda}_1 = -1, \quad \tilde{\boldsymbol{w}}_2 = \frac{1}{\sqrt{2}}[1,-1]^\top, \tilde{\lambda}_2 = 1. \tag{10}$$

This is contrast with (8) and the averaging $\tilde{\boldsymbol{w}}_1$ is associated with $\tilde{\lambda}_1 = -1$ requiring absolute value $|\tilde{\boldsymbol{w}}_1^\top \boldsymbol{d}|$ for invariance. In our LMS descriptor that is implemented on $3\times3\times3$ grids, the flipping transformation does not swap the features on horizontally middle grids but only inverts the sign of μ_x.

Algorithm 1 : Learning invariant forms

Input: $\{\boldsymbol{d}_j, \hat{\boldsymbol{d}}_j\}_{j=1}^n$: n descriptor pairs extracted from the original and the horizontally flipped patches.

1: Compute fundamental matrices $\boldsymbol{R}$, $\hat{\boldsymbol{R}}$ and $\boldsymbol{C}$ in (13).
2: $\boldsymbol{R}_0 = \boldsymbol{R}, \hat{\boldsymbol{R}}_0 = \hat{\boldsymbol{R}}, \boldsymbol{C}_0 = \boldsymbol{C}, \boldsymbol{U}_0 = \boldsymbol{I} \in \mathbb{R}^{m \times m}$ (identity matrix).
3: **for** $i = 1$ to m **do**
4: Ensure orthogonality: $\boldsymbol{R} = \boldsymbol{U}_{i-1}^\top \boldsymbol{R}_0 \boldsymbol{U}_{i-1}$, $\hat{\boldsymbol{R}} = \boldsymbol{U}_{i-1}^\top \hat{\boldsymbol{R}}_0 \boldsymbol{U}_{i-1}$, $\boldsymbol{C} = \boldsymbol{U}_{i-1}^\top \boldsymbol{C}_0 \boldsymbol{U}_{i-1}$.
5: Compute the eigenvector $\boldsymbol{w}_i'$ associated with the maximum eigenvalue λ_i by applying CG [7] to (16).
6: Obtain the weight vector by $\boldsymbol{w}_i = \boldsymbol{U}_{i-1} \boldsymbol{w}_i'$.
7: Compute the subspace bases $\boldsymbol{U}_i \in \mathbb{R}^{m \times m-i}$ which is subject to $\boldsymbol{U}_i \perp \boldsymbol{w}_i$ and $\boldsymbol{U}_{i-1} = \boldsymbol{U}_i \cup \boldsymbol{w}_i$.
8: **end for**

Output: $\{\boldsymbol{w}_i, \lambda_i\}_{i=1}^m$: The weight vectors $\boldsymbol{w}$ associated with the correlation coefficients λ to produce the invariant form (5).

In this case, the transformation matrix $\boldsymbol{T}$ is reduced to the scalar $\boldsymbol{T} = -1$ for μ_x and $\boldsymbol{T} = 1$ for the others, obviously resulting in the invariance form of $\{|\mu_x|, \mu_y, \sigma_x, \sigma_y\}$.

Note that the proposed invariant forms retain the norm of descriptors, $\|\mathrm{g}(\boldsymbol{d})\|_2 = \|\boldsymbol{d}\|_2$, since the eigenvectors are orthonormal due to the symmetric $\boldsymbol{T}$.

3.3. Learning descriptor-level invariance

Unlike the hand-crafted descriptors, it is difficult to explicitly describe the transformation of the ConvNet descriptors [37] in advance since the learned ConvNet is not so understandable for us as to estimate the transformation matrix. Consequently, the descriptor-level invariant form using $\boldsymbol{w}$ (Sec. 3.2) can not be analytically derived. Therefore, we propose a method to statistically *learn* the invariant form for the ConvNet descriptors based on data.

Suppose we have a pair of a descriptor and its flipped one $\{\boldsymbol{d}_j, \hat{\boldsymbol{d}}_j\}_{j=1}^n$ where $\hat{\boldsymbol{d}}_j$ is actually computed on the flipped video sequences only in this training. Our goal is to learn the weight vector $\boldsymbol{w}$ in the invariant form (5) from those pairs. The transformation matrix $\boldsymbol{T}$ can be estimated by the following least-squares;

$$\min_{\boldsymbol{T}} \sum_{j=1}^{n} \|\boldsymbol{T}^\top \boldsymbol{d}_j - \hat{\boldsymbol{d}}_j\|_2^2 + \|\boldsymbol{T}^\top \hat{\boldsymbol{d}}_j - \boldsymbol{d}_j\|_2^2, \tag{11}$$

since $\boldsymbol{d}$ and $\hat{\boldsymbol{d}}$ are transformed to each other by $\boldsymbol{T}$. The optimizer $\boldsymbol{T}^*$ satisfies the following condition,

$$(\boldsymbol{R} + \hat{\boldsymbol{R}})\boldsymbol{T}^* = \boldsymbol{C} + \boldsymbol{C}^\top, \tag{12}$$

$$\text{where } \boldsymbol{C} = \sum_{j=1}^{n} \boldsymbol{d}_j \hat{\boldsymbol{d}}_j^\top,\ \boldsymbol{R} = \sum_{j=1}^{n} \boldsymbol{d}_j \boldsymbol{d}_j^\top,\ \hat{\boldsymbol{R}} = \sum_{j=1}^{n} \hat{\boldsymbol{d}}_j \hat{\boldsymbol{d}}_j^\top. \tag{13}$$

Here, we introduce the eigen-decomposition $\boldsymbol{T}^* = \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^{-1}$ to obtain

$$(\boldsymbol{C} + \boldsymbol{C}^\top)\boldsymbol{W} = (\boldsymbol{R} + \hat{\boldsymbol{R}})\boldsymbol{W}\boldsymbol{\Lambda}, \tag{14}$$

which is a generalized eigenvalue problem.

(14) is interpreted from the other viewpoint of cross-correlation between $\boldsymbol{d}$ and $\hat{\boldsymbol{d}}$. Since a scale of the invariant form $\boldsymbol{w}^\top \boldsymbol{d}$ is irrelevant to the invariance quality, it is necessary to consider only the relationship between the paired values $\{\boldsymbol{w}^\top \boldsymbol{d}, \boldsymbol{w}^\top \hat{\boldsymbol{d}}\}$ in disregard of the scale. Such relationship can be measured by the correlation coefficient between two sequences of $\boldsymbol{w}^\top[\boldsymbol{d}_1, \cdots, \boldsymbol{d}_n, \hat{\boldsymbol{d}}_1, \cdots, \hat{\boldsymbol{d}}_n]$ and $\boldsymbol{w}^\top[\hat{\boldsymbol{d}}_1, \cdots, \hat{\boldsymbol{d}}_n, \boldsymbol{d}_1, \cdots, \boldsymbol{d}_n]$, and the weight $\boldsymbol{w}$ is optimized by maximizing it;

$$\max_{\boldsymbol{w}} \frac{\sum_{j=1}^{n} \boldsymbol{w}^\top (\boldsymbol{d}_j \hat{\boldsymbol{d}}_j^\top + \hat{\boldsymbol{d}}_j \boldsymbol{d}_i^\top) \boldsymbol{w}}{\sum_{j=1}^{n} \boldsymbol{w}^\top (\boldsymbol{d}_j \boldsymbol{d}_j^\top + \hat{\boldsymbol{d}}_j \hat{\boldsymbol{d}}_j^\top) \boldsymbol{w}}, \tag{15}$$

which induces the generalized eigenvalue problem,

$$(\boldsymbol{C} + \boldsymbol{C}^\top)\boldsymbol{w} = \lambda(\boldsymbol{R} + \hat{\boldsymbol{R}})\boldsymbol{w}, \tag{16}$$

where the eigenvalue λ corresponds to the (optimized) correlation coefficient satisfying $-1 \leq \lambda \leq 1$ and this is the same problem as (14).

According to the characteristics of the invariant form for the hand-crafted descriptors (Sec. 3.2), we can naturally impose an orthonormality constraint $\boldsymbol{w}_i^\top \boldsymbol{w}_k = \delta_{ik}$ on (14). For solving such an orthogonal generalized eigenvalue problem, we apply the simple method [8] to sequentially compute the eigenvector of the maximum eigenvalue in (16) while keeping the orthonormality constraint; the conjugate gradient (CG) method [7] is efficiently applied to solve (16), of which the detailed algorithm is shown in Alg. 1. The invariant form of the ConvNet descriptors is given by (5) with the optimized $\{\boldsymbol{w}_i, \lambda_i\}_{i=1}^m$; the learned form of $|\lambda| < 1$ does not provide complete invariance, but effectively increases robustness to flipping. It should be noted that the invariant forms presented in Sec. 3.2 for the hand-crafted descriptors can also be retrieved as the optimizer in this formulation.

The proposed optimization problem is composed of only three fundamental matrices, $\boldsymbol{R}$, $\hat{\boldsymbol{R}}$ and $\boldsymbol{C}$ in (13), which are computed efficiently. Due to the CG-based optimization method [7], Alg. 1 can be efficiently applied even to the high dimensional vectors. As the pair-wise correspondence is embedded into the cross-correlation matrix $\boldsymbol{C}$, we can efficiently leverage a large number of local descriptors to construct the optimization problem without retaining each of samples; in the experiments (Sec. 4.2), *all* the local descriptors extracted from the training videos are used for learning.

On the hand-crafted descriptors, the descriptor-level invariance is efficiently achieved via (5) only by adding or subtracting two components (8,10); the computation cost is negligible as in the case of patch-level invariance (Sec. 3.1). Even for the ConvNet descriptors [37], (5) is efficiently computed by sophisticated matrix-vector routine, such as `gemv` in BLAS library. It is noteworthy that the proposed method to render descriptor-level invariance is so general as to be applicable to versatile descriptors/variations, not limited to motion descriptors; it is our future work to show the portability of the method.

4. Experimental results

We apply the proposed methods to render flip invariance to local motion descriptors on action classification tasks. The methods are summarized in Table 2 and the detailed procedures are shown in the supplemental material; note again that they operate on the descriptors with quite a low computation cost as post-processing, except for the case of ConvNet descriptors [37] with patch-level invariance (Sec. 3.1) which requires re-computing descriptors.

An input video sequence is represented in the bag-of-features framework which encodes the local motion descriptors by means of the improved Fisher kernel (iFK) method [27], and then a linear classifier trained by SVM [34] is applied to finally categorize the video descriptor (iFK feature vector) into the action classes. The local descriptors (Sec. 2) are extracted on the short-term trajectories densely sampled from the video sequence [36]. Note that all these descriptors, even including the ConvNet ones [37], are computed at the same dense trajectories, in order to make fair comparison across descriptors; we directly employ the method [37] to compute ConvNet descriptors, and for the detail, refer to [37]. Based on the preliminary study, the PCA projection of local descriptors into 64-dimensional subspace with 256 GMMs is employed to produce favorable performance in terms both of the classification accuracy and the computation cost as reported in [37]. Through the fixed classification pipeline, we can quantitatively evaluate the performance of local descriptors themselves on three datasets; HMDB51 [18], UCF101 [30] and Hollywood2 [23]. The classification performance is measured according to the standard protocols provided with the respective datasets.

4.1. Hand-crafted local descriptors

As described in Sec. 2, we employ the commonly used motion descriptors, HOG [5], HOF [20] and MBH$_{x/y}$ [6] as well as LMS presented in this paper. The performance results are shown in Table 3. We can see in Table 3i that in the original setting, LMS works well, being comparable to HOF and MBH$_y$ with superiority over HOG, in spite of its simple formulation which contributes to fast computation. And, the common technique augmenting training and test samples by flipping[1] less contributes to performance improvement.

Then, we investigate the proposed method of patch-level invariance (Sec. 3.1). The hand-crafted descriptors are efficiently flipped by analytically transforming them via the matrix $\boldsymbol{T}$ (Sec. 2.1&3.2) if the patch-level invariance method suggests to flip the patch. The proposed flow-based method (Flip-by-Flow) is compared with the previous one [43] which is proposed in the image domain exploiting the image gradients. As shown in Table 3ii, the proposed method is superior to the previous one, demonstrating that motion flow is useful for estimating the patch orientation. This is because the flows are more robust and stable compared to the image gradients as discussed in Sec. 3.1. To empirically show the stability, we depict in Fig. 2a the entropy of the distribution of flow and image gradient orientations on the patch by utilizing HOF and HOG of 8 orientation bins. The image gradients exhibit high entropy meaning diverse gradient orientations, which causes unstable estimation of the dominant patch orientation. In contrast, the flows are of lower entropy, which contributes to the stable estimation. The alternative method (Flip-by-Trajectory) leverages relatively longer-term flow information to estimate the dominant orientation compared with Flip-by-Flow. Though they work similarly outperforming the gradient-based method [43], Flip-by-Trajectory is slightly inferior to Flip-by-Flow. The Flip-by-Flow aggregates the flows distributed around the trajectory while the Flip-by-Trajectory focuses only on the tracked points causing less stability. It is noteworthy that the proposed method equips the HOG de-

Table 3. Performance (%) comparison on hand-crafted descriptors. The proposed methods are highlighted in bold.

(a) Hollywood2 [23]

	Method	HOG	HOF	LMS	MBH$_x$	MBH$_y$
i	original	47.62	58.16	60.78	54.43	58.61
	data augmentation	47.80	58.07	59.89	54.51	59.58
ii	Flip by Gradient [43]	48.15	60.24	61.51	55.36	60.25
	Flip by Trajectory	50.30	60.50	60.41	57.31	61.60
	Flip by Flow	**50.37**	**60.42**	**61.86**	**55.92**	**61.76**
iii	Invariant by Avg.	48.06	60.24	61.26	53.06	58.31
	Invariant by Max	48.46	60.51	60.17	52.27	57.99
	MI-SIFT [22]	48.29	60.63	62.30	52.77	58.94
	Invariant by Eig. (5)	**48.45**	**61.55**	**62.12**	**55.50**	**60.24**

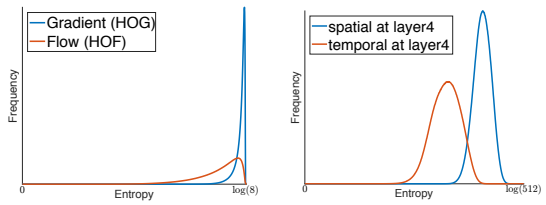
(b) HMDB51 [18]

	Method	HOG	HOF	LMS	MBH$_x$	MBH$_y$
i	original	43.07	50.00	49.72	43.94	50.35
	data augmentation	42.70	49.59	49.35	43.27	50.35
ii	Flip by Gradient [43]	44.25	52.24	50.94	46.34	50.74
	Flip by Trajectory	45.27	52.68	52.11	46.49	51.79
	Flip by Flow	**45.49**	**52.81**	**51.87**	**47.12**	**51.90**
iii	Invariant by Avg.	43.16	52.24	50.85	44.10	48.00
	Invariant by Max	42.14	52.05	48.65	42.72	47.02
	MI-SIFT [22]	43.42	52.09	49.93	44.34	49.11
	Invariant by Eig. (5)	**44.31**	**54.64**	**51.72**	**45.21**	**50.54**

(c) UCF101 [30]

	Method	HOG	HOF	LMS	MBH$_x$	MBH$_y$
i	original	73.81	77.44	78.70	75.64	77.88
	data augmentation	74.18	77.69	78.34	75.43	77.79
ii	Flip by Gradient [43]	75.38	78.47	79.58	77.59	78.92
	Flip by Trajectory	75.79	79.04	80.36	78.26	79.58
	Flip by Flow	**75.47**	**79.27**	**80.48**	**78.17**	**79.58**
iii	Invariant by Avg.	73.67	77.08	79.12	74.38	74.64
	Invariant by Max	72.99	77.41	78.07	72.86	73.88
	MI-SIFT [22]	74.54	77.24	79.31	75.50	75.64
	Invariant by Eig. (5)	**74.77**	**80.09**	**79.69**	**76.17**	**78.02**

[1] In the test phase, we average the two classification scores of the original and flipped videos to achieve flip invariance.



(a) HOF vs. HOG (b) ConvNet descriptors

Figure 2. Comparison of entropy in orientation histograms of flows (HOF) and image gradients (HOG) (a), and in the spatial and temporal ConvNet descriptors (b) on HMDB51 dataset.

scriptor with flow information; that is, even the same HOG descriptors can be distinguished by patch-flipping which reflects the horizontal motion direction estimated by the proposed Flip-by-Flow method.

As to the descriptor-level invariance (Sec. 3.2), the proposed method is compared with the simple methods of component-wise operators including averaging $\frac{\boldsymbol{d}+\hat{\boldsymbol{d}}}{2}$ and maximizing $\max(\boldsymbol{d}, \hat{\boldsymbol{d}})$ as well as MI-SIFT [22] which is the operator concatenating $\boldsymbol{d}+\hat{\boldsymbol{d}}$ and $\boldsymbol{d}\circ\hat{\boldsymbol{d}}$ (component-wise product). The proposed method outperforms the others as shown in Table 3iii. The method exploits the difference between the feature components in addition to their average so that the invariant form retains the information (descriptive power) of the descriptors as much as possible, which is theoretically derived from the transformation matrix.

Comparing the two types of proposed methods, the patch-level invariance based on flows works slightly better on the MBH descriptors, while the invariant form (5) favorably improves the HOF and LMS descriptors. The MBH descriptors are based on the derivatives of flows resulting in a somewhat complicated (diverse) histogram features, while the HOF and LMS derived from the raw flows would produce the simpler features of low entropy as shown in Fig. 2a. The invariant form (5) of simple transformation comprising averaging and differencing is suitable to characterize the simple descriptors rather than the complicated ones which might require finer treatment to extract the effective information; the simple averaging and differencing are enough to characterize the features of low entropy in which the feature values are concentrated on a few bins.

4.2. ConvNet local descriptors

As suggested in [37], we apply the spatial ConvNet with layer 4&5 and the temporal ConvNet with layer 3&4 to extract local motion descriptors; refer to Table 1 of [37] for the detailed architectures of the ConvNets. In this case, the path-level invariance methods (Sec. 3.1) and the methods of averaging, maximizing and MI-SIFT [22] double the ConvNet computation due to re-computing the flipped descriptors, while the learned invariant form (5) (Sec. 3.3) works solely on the original ConvNet descriptors without considering their flipped ones, which keeps the computation cost almost the same as the original process. We here categorize the methods from this computational viewpoint and discuss performance results in Table 4. Note that the invariant form is learned by applying Alg. 1 to all the ConvNet descriptors found in the training set for respective datasets.

Table 4. Performance (%) comparison on ConvNet descriptors [37]. The proposed methods are highlighted in bold.

(a) Hollywood2 [23]

	Method	spatial layer4	spatial layer5	temporal layer3	temporal layer4
i	original	48.90	49.42	62.72	64.20
	data augmentation	49.38	49.53	62.51	65.34
ii	Flip by Gradient [43]	49.66	50.00	64.27	66.20
	Flip by Flow	**51.59**	**50.01**	**65.08**	**65.96**
	Invariant by Avg.	49.50	49.58	64.24	66.27
	Invariant by Max	49.83	51.75	65.54	66.02
	MI-SIFT [22]	48.41	45.89	60.50	60.92
iii	Invariant by Learn	**48.98**	**49.45**	**64.06**	**65.46**
	Invariant by Learn (gEig)	48.85	48.78	65.12	64.93
	Invariant by Learn (UCF101)	49.05	49.50	63.82	65.56

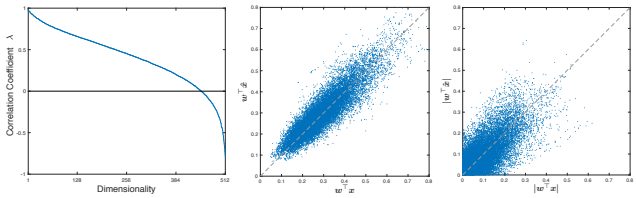
(b) HMDB51 [18]

	Method	spatial layer4	spatial layer5	temporal layer3	temporal layer4
i	original	46.67	45.16	54.16	56.45
	data augmentation	45.88	45.77	54.36	57.49
ii	Flip by Gradient [43]	46.12	45.12	56.01	58.28
	Flip by Flow	**47.21**	**45.58**	**56.25**	**58.52**
	Invariant by Avg.	45.73	45.01	54.66	57.49
	Invariant by Max	47.43	45.73	56.30	58.26
	MI-SIFT [22]	45.34	42.88	50.89	54.34
iii	Invariant by Learn	**46.41**	**45.32**	**54.66**	**57.30**
	Invariant by Learn (gEig)	44.66	44.44	53.70	56.45
	Invariant by Learn (UCF101)	46.62	45.38	54.90	56.80

(c) UCF101 [30]

	Method	spatial layer4	spatial layer5	temporal layer3	temporal layer4
i	original	78.19	75.84	79.91	83.19
	data augmentation	78.63	76.18	80.30	83.86
ii	Flip by Gradient [43]	78.46	75.52	81.52	84.50
	Flip by Flow	**79.15**	**75.53**	**82.11**	**84.62**
	Invariant by Avg.	78.35	75.69	80.68	84.47
	Invariant by Max	78.68	76.44	81.71	85.14
	MI-SIFT [22]	77.13	71.69	77.70	82.51
iii	Invariant by Learn	**78.43**	**75.73**	**80.75**	**83.81**
	Invariant by Learn (gEig)	76.41	74.48	80.33	83.70

As shown in Table 4ii, the patch-level invariance method based on flows favorably improves performance in comparison to that based on gradients [43] as is the case with the hand-crafted descriptors (Table 3ii). For these ConvNet descriptors, the operator of maximizing is superior to averaging and MI-SIFT [22], producing similar performance to the flow-based method. The features produced by the ConvNet exhibit existence of certain types of object in a moderately higher semantic level and thus those features would be well enhanced by the maximizing operator which is also com-



(a) λ (b) $\lambda = 0.9872$ (c) $\lambda = -0.7832$

Figure 3. Learned invariant form for the temporal ConvNet descriptors of layer4 on UCF101 (a), and its application to descriptors in HMDB51 (b).

patible with the max-pooling embedded in the ConvNet.

Table 4iii shows that the learned invariant form (Sec. 3.3) can also improve performance especially on the temporal ConvNet descriptors. As in the comparison of image gradients and flows in Fig. 2a, Fig. 2b shows that the temporal ConvNet descriptors are simpler and of lower entropy compared to those of the spatial ConvNet. Thus, the similar discussion with Sec. 4.1 holds for the reason why the proposed invariant form works on the temporal ConvNet descriptors better than on the spatial ones. The obtained eigenvalues λ (correlation coefficients) are shown in Fig. 3a; there are negative eigenvalues on which the absolute operator is applied to encourage positive correlation and the examples of the projection by the invariant form are shown in Fig. 3bc. In case of removing orthonormality constraints from the invariant form, our optimization problem results in just a generalized eigenvalue problem (16) which can be solved by the off-the-shelf solver. The performance of so optimized invariant form is also shown in Table 4iii. It, however, is inferior to the proposed one, and thus we can conclude that the orthonormality constraint is effective even for the ConvNet descriptors not only for the hand-crafted descriptors.

We also show generality of the learned invariant form across the datasets in Table 4iii. The invariant form learned on UCF101 dataset works similarly to those learned on the respective datasets; we here employ UCF101 as it is the largest among the three datasets in our experiments. The descriptors characterize spatio-temporally local motions, not a whole video, and thus it is less sensitive to the environments where the video is captured. Therefore, the invariant form learned from a large number of the local descriptors are general across the datasets and transferable to the other datasets than the one used in learning. This result suggests that the invariant form learned such as on UCF101 is applicable to versatile datasets for action recognition without re-learning it on each dataset.

4.3. Combination of descriptors

Finally, we show the classification performance by combining descriptors. In this experiment, we concatenate the iFK feature vectors each of which is computed by using each type of descriptor. We consider two scenarios in the combination; the first one is oriented toward computation efficiency by the proposed invariant form (Sec. 3.2&3.3) and the other is for high classification accuracy by the proposed patch-level invariance method (Sec. 3.1). The performance results are shown in Table 5. As to the hand-crafted descriptors, those two methods produce comparable performance, successfully improving the original one. In the combination with the ConvNet descriptors, the proposed methods favorably improve performance being comparable to the state-of-the-arts; 68.0% [19] and 73.6% [13] on Hollywood2, 65.1% [19] and 66.8% [25] on HMDB51, and 88.0% [29], 89.1% [19] and 90.4% [32] on UCF101.

Table 5. Combination of descriptors. 'iDT' means the descriptor set of HOG+HOF+MBH$_x$+MBH$_y$ [36], while 'ConvNet' indicates the set of all (four) ConvNet descriptors.

(a) Hollywood2

Descriptor	orig.	Invariant by Eig./Learn	Flip by Flow
iDT [36]	64.25	66.27	65.70
iDT+LMS	64.94	66.64	66.31
ConvNet [37]	68.94	69.92	70.93
iDT+LMS+ConvNet	70.53	72.12	72.49

(b) HMDB51

Descriptor	orig.	Invariant by Eig./Learn	Flip by Flow
iDT [36]	58.00	60.11	59.98
iDT+LMS	58.69	60.22	60.20
ConvNet [37]	62.79	63.22	63.86
iDT+LMS+ConvNet	64.38	65.14	65.71

(c) UCF101

Descriptor	orig.	Invariant by Eig./Learn	Flip by Flow
iDT [36]	85.32	86.29	86.36
iDT+LMS	85.04	86.20	86.38
ConvNet [37]	88.74	89.16	89.54
iDT+LMS+ConvNet	89.66	90.27	90.63

5. Conclusion

We have proposed methods to render flip invariance for local motion descriptors via two approaches. One is to make the input patch flip-invariant by efficiently estimating a patch orientation based on flows, and thereby the descriptor extracted from the invariant patch is invariant to flipping. The other method derives the invariant form from the transformation (matrix), which is explicitly obtained in the hand-crafted descriptors. It is also extended to learn the invariant form for the ConvNet descriptors whose flip transformation is not explicitly given. The latter method is advantageous in terms of computational cost since it works solely on the original descriptors without re-computing the flipped ones. In the experiments on action classification, the proposed methods favorably improve the performance with superiority to the others invariant methods.

References

[1] B. Beauzamy. *Introduction to Operator Theory and Invariant Subspaces*. North Holland, 1988.

[2] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, pages 321–328, 2013.

[3] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: An evaluation of recent feature encoding methods. In *BMVC*, 2011.

[4] S. Clinchant, G. Csurka, F. Perronnin, and J. Renders. XRCE's Participation to ImageEval. In *ImageEval workshop at CVIR*, 2007.

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

[6] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, pages 428–441, 2006.

[7] Y. Feng and D. Owen. Conjugate gradient methods for solving the smallest eigenpair of large symmetric eigenvalue problems. *International Journal For Numerical Methods in Engineering*, 39(13):2209–2229, 1996.

[8] D. H. Foley and J. W. Sammon. An optimal set of discriminant vectors. *IEEE Transactions on Computers*, 24(3):281–289, 1975.

[9] X. Guo and X. Cao. Find: A neat flip invariant descriptor. In *ICPR*, pages 515–518, 2010.

[10] X. Guo and X. Cao. Mift: A framework for feature descriptors to be mirror reflection invariant. *Image and Vision Computing*, 30(8):546–556, 2012.

[11] T. Hassner. A critical review of action recognition benchmarks. In *CVPR Workshop*, pages 245–250, 2013.

[12] N. J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[13] M. Hoai and A. Zisserman. Improving human action recognition using score distribution and ranking. In *ACCV*, 2014.

[14] L. Isik, A. Tacchetti, and T. Poggio. Fast, invariant representation for human action in the visual system. *arXiv,* 1601.01358, 2016.

[15] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, pages 2555–2562, 2013.

[16] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[18] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011.

[19] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In *CVPR*, 2015.

[20] I. Laptev, M. Marzalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

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

[22] R. Ma, J. Chen, and Z. Su. Mi-sift: Mirror and inversion invariant generalization for sift descriptor. In *CIVR*, pages 228–236, 2010.

[23] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, pages 2929–2936, 2009.

[24] V. Parameswaran and R. Chellappa. View invariants for human action recognition. In *CVPR*, pages 613–619, 2003.

[25] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *ECCV*, pages 581–595, 2014.

[26] A. D. Polyanin and A. V. Manzhirov. *Handbook Of Mathematics For Engineers and Scientists*. Chapman and Hall/CRC, 2006.

[27] 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.

[28] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, pages 32–36, 2004.

[29] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[30] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[31] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[33] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

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

[35] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103:60–79, 2013.

[36] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013.

[37] L. Wang and Y. Qiao. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015.

[38] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016.

[39] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2):249–257, 2006.

[40] A. D. Wilson and A. F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:884–900, 1999.

[41] L. Xia, C.-C. CHen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPR*, pages 20–27, 2012.

[42] L. Xie, Q. Tian, and B. Zhang. Max-sift: Flipping invariant descriptors for web logo search. In *ICIP*, pages 5716–5720, 2014.

[43] L. Xie, J. Wang, W. Lin, B. Zhang, and Q. Tian. Ride: Reversal invariant descriptor enhancement. In *ICCV*, pages 100–108, 2015.

[44] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. *arXiv,* 1502.08029, 2015.

[45] W.-L. Zhao and C.-W. Ngo. Flip-invariant sift for copy and object detection. *IEEE Transactions on Image Processing*, 22(3):980–991, 2013.