Flip-Invariant SIFT for Copy and Object Detection

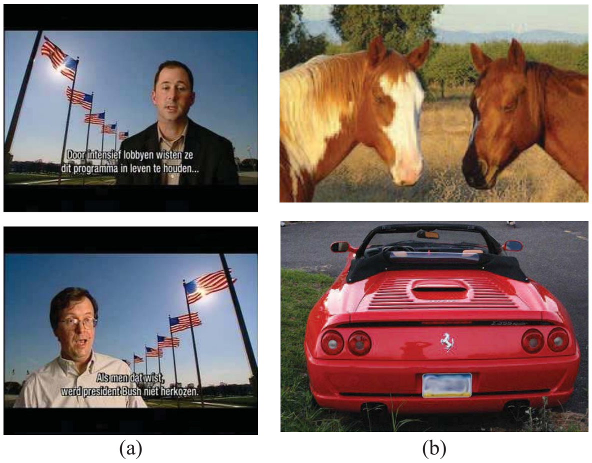
Wan-Lei Zhao and Chong-Wah Ngo, *Member, IEEE*

***Abstract*— Scale-invariant feature transform (SIFT) feature has been widely accepted as an effective local keypoint descriptor for its invariance to rotation, scale, and lighting changes in images. However, it is also well known that SIFT, which is derived from directionally sensitive gradient fields, is not flip invariant. In real-world applications, flip or flip-like transformations are commonly observed in images due to artificial flipping, opposite capturing viewpoint, or symmetric patterns of objects. This paper proposes a new descriptor, named flip-invariant SIFT (or F-SIFT), that preserves the original properties of SIFT while being tolerant to flips. F-SIFT starts by estimating the dominant curl of a local patch and then geometrically normalizes the patch by flipping before the computation of SIFT. We demonstrate the power of F-SIFT on three tasks: large-scale video copy detection, object recognition, and detection. In copy detection, a framework, which smartly indices the flip properties of F-SIFT for rapid filtering and weak geometric checking, is proposed. F-SIFT not only significantly improves the detection accuracy of SIFT, but also leads to a more than 50% savings in computational cost. In object recognition, we demonstrate the superiority of F-SIFT in dealing with flip transformation by comparing it to seven other descriptors. In object detection, we further show the ability of F-SIFT in describing symmetric objects. Consistent improvement across different kinds of keypoint detectors is observed for F-SIFT over the original SIFT.**

***Index Terms*— Flip invariant scale-invariant feature transform (SIFT), geometric verification, object detection, video copy detection.**

I. Introduction

DUE TO the success of SIFT [1], image local features have been extensively employed in a variety of computer vision and image processing applications. Particularly, various recent works take advantage of SIFT to develop advanced object classifiers. The studies conducted by [2], [3], for example, show that using aggregated local features based on SIFT, the performance of linear classifier is comparable to more sophisticated but computationally expensive classifiers. The attractiveness of SIFT is mainly due to its invariance to various image transformations including: rotation, scaling, lighting changes and displacements of pixels in a local region. SIFT is normally computed over a local salient region which is located by multi-scale detection and rotated to its dominant orientation. As a result, the descriptor is invariant to both scale and rotation. Furthermore, due to spatial partitioning and 2D directional gradient binning, SIFT is insensitive to color, lighting and small pixel displacement. Despite these desirable properties, SIFT is not flip invariant. As a consequence, the descriptors extracted from two identical but flipped local patches could be completely different in feature space. This has degraded the effectiveness of feature point matching [4] and introduced extra computational overhead [5]–[7] for applications such as video copy detection.


Door intensief lobbyen wisten ze
dit programma in leven te houden...
Als men dat wist,
werd president Bush niet herkozen.

(a) (b)

Fig. 1. Examples of flipping in different contexts. (a) Viewpoint change. (b) Flip-like structure.

Flip or flip-like operations happen in different contexts. In copyright infringement, flip operation has been one of the frequently used tricks [8], [9]. Especially, horizontal flipping is more commonly observed since this operation visually will not result in any apparent loss of image/video content. Flips also occur when taking pictures of a scene from opposite viewpoints. This kind of flips, as shown in Figure 1(a), is usually captured in different snapshots of time, and widely exists especially in TV news programs broadcast by different channels. In addition, objects having symmetric structure also exhibit flip-like transformation as shown in Figure 1(b). Generally speaking, allowing the symmetric structure of objects to be matched in the feature space will increase the chance of recalling objects in the same classes, especially when the objects are captured from arbitrary viewpoints. In short, the ability of a descriptor in characterizing the visual invariance of a local region despite of whether the region is flipped or inherently symmetric is important for tasks such as copy and object detection.

In the literature, there are several local descriptors such as SPIN [10] and RIFT [10] which are flip invariant.

Manuscript received August 25, 2011; revised February 25, 2012; accepted September 25, 2012. Date of publication October 22, 2012; date of current version January 24, 2013. This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 119610). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kiyoharu Aizawa.

W.-L. Zhao is with INRIA-Rennes, Rennes Cedex 35042, France, and also with the Department of Computer Science, University of Kaiserslautern, Kaiserslautern 67663, Germany (e-mail: wanlei.zhao@inria.fr).

C.-W. Ngo is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: cwngo@cs.cityu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

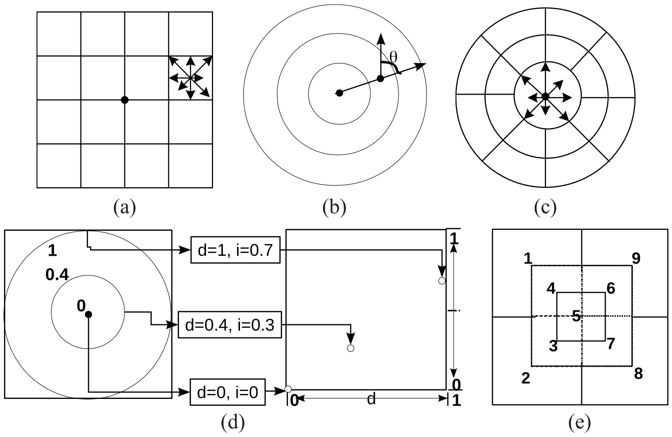
Digital Object Identifier 10.1109/TIP.2012.2226043

However, both descriptors are sensitive to scale changes, and as reported in [11], are not as discriminative as SIFT. In contrast, this paper proposes F-SIFT which enriches SIFT with flip invariant property while preserving its feature distinctiveness. By observing that flip operation with respect to arbitrary axis can be decomposed into a horizontal (or vertical) flip followed by rotation, F-SIFT first computes the dominant curl of gradient fields in a local patch. The curl classifies a patch into either clockwise or anti-clockwise, and F-SIFT explicitly flips a patch of anti-clockwise before extracting SIFT feature. Intuitively, flip invariance is achieved by geometrically normalizing local patch before the computation of SIFT.

The main contribution of this paper is the proposal of F-SIFT which enhances SIFT with flip invariance property. The employment of F-SIFT for video copy detection, object recognition and detection is also demonstrated. Particularly, we show that, by smartly indexing F-SIFT, the performance improvement in both detection accuracy and speed could generally be expected. The remaining of this paper is organized as follows. Section II reviews variants of local descriptors and their utilization for copy and object detection. Section III describes the extraction of F-SIFT descriptors from local regions. Section IV further presents a framework for large-scale video copy detection, by proposing the schemes for feature indexing and weak geometric checking based on F-SIFT. Section V presents a comparative study to investigate the effect of detectors and descriptors in face of flip transformation for object recognition. Section VI empirically compares the performance of F-SIFT and SIFT for object detection. Finally Section VII concludes this paper.

II. Related Work

While developing local descriptors invariant to various geometric transformations has received numerous research attention, the property of flip invariance surprisingly is often not considered. Until recently, there are several flip invariant descriptors including RIFT [10], SPIN [10], MI-SIFT [12] and FIND [13]. These descriptors, including SIFT, mainly differ by the partitioning scheme of local region as shown in Figure 2. SIFT, which divides a region into 4 × 4 blocks and describes each grid with an 8 directional gradient histogram as in Figure 2(a), generates the feature by concatenating the histograms in row major order from left to right and the histogram bins in clockwise manner. As a result, flip transformation of the region will disorder the placement of blocks and bins. This results in a different version of descriptor due to the predefined order of feature scanning. The potential solutions for dealing with this problem include altering the partitioning scheme or scanning order [10], [13], and feature transformation [12].



(a) (b) (c)

(d) (e)

Fig. 2. Partition schemes of (a) SIFT [1], (b) RIFT [10], (c) GLOH [11], (d) SPIN [10], and (e) FIND [13].

RIFT [10] adopts a different partitioning scheme than SIFT by dividing a region along the log-polar direction as shown in Figure 2(b). Similar to SIFT, the 8-directional histograms are computed for each division and then concatenated to form a descriptor. Since the partitioning scheme itself is flip and rotation invariant, RIFT is not sensitive to order of scanning. On the other hand, while this radius based division is smooth and less vulnerable to quantization loss if compared to grid-based partitioning, the spatially loose representation also results in RIFT a descriptor not as distinctive as SIFT. GLOH which can be viewed as an integrated version of SIFT and RIFT provides finer partitioning as shown in Figure 2(c). However, the invariance property no longer exists once after strengthening the spatial constraint. SPIN as shown in Figure 2(d), instead, preserves flip invariance property while enforcing spatial information by encoding a region as a 2D histogram of pixel intensity and distance from region center. Despite the improvement, nevertheless, the empirical evaluation in [11] reported that SPIN as well as RIFT and GLOH are outperformed by SIFT.

FIND [13] is a new descriptor which allows overlapped partitioning and scans the 8-directional gradient histograms by following the order indicated in Figure 2(e). Under this scheme, the descriptors produced before and after a flip operation are also mirror of each other. Specifically, a descriptor generated as a result of flip can be recovered by scanning the histograms in reverse order. With this interesting property, FIND explicitly makes the descriptor invariant to flip by estimating whether a region is left or right pointing through parameter thresholding. When comparing two descriptors of left and right pointing respectively, the descriptor components are rearranged on the fly for proper order of feature matching. Nevertheless, as reported in [13], the estimation of pointing direction is highly dependent on parameter setting, and more importantly, incorrect estimation directly implies invalid matching result. In addition, similar to RIFT, the partitioning scheme does not produce descriptor as distinctive as SIFT. MI-SIFT [12], instead, operates directly on SIFT while transforming it to a new descriptor which is flip invariant. This is achieved by explicitly identifying the groups of feature components which are disorderly placed as a result of flip operation. MI-SIFT labels 32 of such groups and represents each group with four moments which are flip invariant. Nevertheless, the descriptor based on moment is not discriminative. As reported in [12], this results in more than *10%* of matching performance degradation than SIFT when no-flip transformation happens.

Flip operations have been viewed as one of the widely used infringement tricks. In TRECVID copy detection task (CCD) [9], [14], for instance, video copies as a result of flips are regarded as one of the major testing items. Interestingly, most participants in CCD nevertheless seldom adopted flip invariant descriptors, and instead, employed SIFT for its feature distinctiveness. The problem of flipped copy detection is engineered by indexing two SIFT descriptors for each region [6], [15], of which one of them is computed by simulating flip operation. This results in significant increase in both indexing time and memory consumptions. In [5], [7], an alternative strategy was employed by submitting two versions of descriptors, flipped and without flipped, as query for copy detection. This strategy introduces the drawback that the query processing time is double.

Most of the keypoint detectors and visual descriptors are proposed for feature point matching in object recognition [11]. However, there is no systematic and comparative studies yet to investigate their performance in face of flip transformation. Different from copy detection and object recognition, the existing works on object detection are mostly learning-based. Specifically, bag-of-visual-words (BoW) constructed from local features such as SIFT are input for classifier learning [16], [17]. To the best of our knowledge, no work has yet seriously addressed the issue of detection performance by contrasting features with and without incorporating flip invariance property.

III. Flip Invariant SIFT

We begin by describing the existing salient region (or keypoint) detectors. These detectors are indeed flip invariant and capable of locating regions under various transformations. In other words, the problem arisen as result of flip operations is originated from the feature descriptor itself. With this fact, we will then present our proposed descriptor F-SIFT which revises SIFT to be flip invariant.

A. Flip Invariant Detectors

There are various keypoint detectors available in the literature [1], [18]–[20]. In general, these detectors perform scale-space analysis for locating local extremes of an image in the selected scales. The outputs are salient points, each associated with a region of support and its dominant orientation. Detectors are mostly similar to each other except with variation in the choice of saliency function. Analysis on flip invariance of major detectors is given as follows.

Given a pixel P, the second moment matrix is defined to describe gradient distribution in the local neighborhood of P:

$$\mu(P,\sigma_I,\sigma_D)=\sigma_D^2 g(\sigma_I)*\begin{bmatrix} L_x^2(P,\sigma_D) & L_xL_y(P,\sigma_D)\\ L_xL_y(P,\sigma_D) & L_y^2(P,\sigma_D)\end{bmatrix} \tag{1}$$

where σ_I is the integration scale, σ_D is the differential scale and L_g is to compute the derivative of P in g (x or y) direction. The local derivatives are computed with Gaussian kernels of the size determined by the scale σ_D. The derivatives are averaged in the neighborhood of P by smoothing with integration scale σ_I. Based on Eqn. 1, the Harris function at pixel P is given by

$$\mathcal{H}arris(P)=Det(\mu(P,\sigma_I,\sigma_D))-\alpha \times Trace^2(\mu(P,\sigma_I,\sigma_D)) \tag{2}$$

where α is a constant. Scale invariance is further achieved by scale-space processing computed by Laplacian-of-Gaussian matrix

$$LoG(P,\sigma_I)=\sigma_I^2|L_{xx}(P,\sigma_I)+L_{yy}(P,\sigma_I)| \tag{3}$$

where L_{gg} denotes the second order derivative in direction g. The local maxima value of P, with respect to an integration scale σ_I, is determined based on the characteristic structure around P. Harris-Laplacian (HarLap) detector regards a pixel P as keypoint if it attains local maxima in $\mathcal{H}arris(P)$ and $LoG(P,\sigma_I)$ simultaneously.

Eqn. 1 involves the computation of the first order derivatives which are directionally sensitive. A horizontal flip transformation, for example, will reverse the sign of derivative along x direction. Fortunately, the second moment matrix is symmetric and the derivatives are squared, resulting in no change of effect on the resulting determinant. While for Eqn. 3, the computation fully relies on the second order derivatives along x and y directions which is typically in following form

$$L_{gg}(P,\sigma_I)=I(g-1,\sigma_I)+I(g+1,\sigma_I)-2*I(g,\sigma_I). \tag{4}$$

Since the Gaussian window is isotropic, L_{gg} remains unchanged in each direction. As a result, flip produces no effect on Eqn. 3. HarLap and Laplacian-of-Gaussian (LoG) are detectors that adopt Eqn. 3 as saliency function.

Difference-of-Gaussian (DoG) detector [1] defines local extrema in spatial and scale spaces based on following function:

$$DoG(P,\sigma)=G(P,k\cdot\sigma)-G(P,\sigma) \tag{5}$$

where $G(P,\sigma)$ is the Gaussian blur applied on pixel P and k is a constant multiplicative factor. Similar to HarLap and LoG, flip operation will take no effect on Eqn. 5 due to the isotropic Gaussian window. Thus, DoG detector is also flip invariant.

Hessian detector, instead, defines the saliency function solely based on the determinant of Hessian matrix as following:

$$\mathcal{H}essian(P,\sigma_D)=\begin{bmatrix} L_{xx}(P,\sigma_D) & L_{xy}(P,\sigma_D)\\ L_{yx}(P,\sigma_D) & L_{yy}(P,\sigma_D)\end{bmatrix}. \tag{6}$$

Flip operation makes no effect on either L_{xx} or L_{yy} but swaps L_{xy} and L_{yx} in the matrix. However, because saliency is computed based on determinant, the swapping will not result in change of value and thus the detector is also flip invariant. Similar analysis applies to Fast Hessian (FastHess) [20] detector. Meanwhile, it is also easy to see that Hessian-Laplacian (HessLap) detector which is defined on Eqn. 3 and Eqn. 6 is also flip invariant.

B. F-SIFT Descriptor

While keypoint detectors are mostly flip invariant, there is no guarantee that the features extracted from salient regions are also flip invariant. As discussed in Section II, the invariance is mainly dependent on the layout of partitioning scheme in a descriptor. Different from the existing approaches, our aim here is to enrich SIFT to be flip invariant while preserving its original properties including the grid-based quantization.

Flip transformation can happen along arbitrary axis. However, it is easy to imagine that any flip can be decomposed into as a flip along a predefined axis followed by a certain degree of rotation as shown in Figure 3. Thus, an intuitive idea to make a descriptor flip invariant is by normalizing a local region before feature extraction through rotating the region to a predefined axis and then flipping it along the axis. Furthermore, if a region has been rotated to its dominant orientation which is the case for regions identified by keypoint detectors, the normalization can be simply done by flipping the region horizontally (or vertically). In other words, a prominent solution for flip invariance is to determine whether flip should be performed before extracting local feature from the region.

We propose dominant curl computation to answer this question. Curl [21] is mathematically defined as a vector operator that describes the infinitesimal rotation of a vector field. The direction of curl is the axis of rotation determined by the right-hand rule. In multivariate calculus, given a vector field $F(x, y, z)$ defined in R^3 which is differentiable in a region, the curl of F is given by

$$\nabla \times F = \begin{vmatrix} i & j & k \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ F_1 & F_2 & F_3 \end{vmatrix}. \tag{7}$$

According to Stokes' theorem, the integration of curl in a vector field can be expressed by

$$\iint_{\Sigma \in R^3} = \nabla \times F \cdot d\Sigma. \tag{8}$$

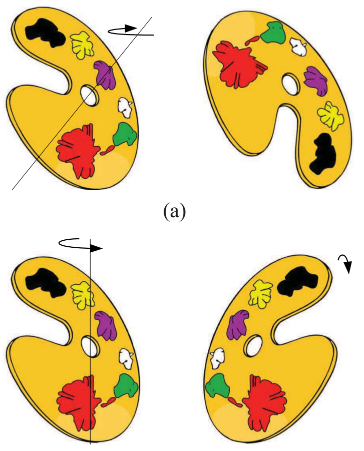
In our case, curl is defined in a 2D discrete vector field I. The curl at a point is the cross product on the first order partial derivatives along x and y directions respectively. The flow (or dominant curl) along the tangent direction can be defined by

$$C = \sum_{(x,y)\in I} \sqrt{\frac{\partial I(x,y)}{\partial x}^2 + \frac{\partial I(x,y)}{\partial y}^2} \times \cos\theta \tag{9}$$

where

$$\frac{\partial I(x,y)}{\partial x} = I(x-1,y) - I(x+1,y)$$
$$\frac{\partial I(x,y)}{\partial y} = I(x,y-1) - I(x,y+1)$$

and θ is the angle from direction of the gradient vector to the tangent of the circle passing through (x, y).


(a)

(b)

Fig. 3. Standardizing arbitrary flip (a) to as a horizontal flip followed by (b) rotation.

Generally, there are only two possible directions for C, either clockwise or counter clockwise, which is indicated by its sign. The sign changes only when the vector field has been flipped (along an arbitrary axis). If we enforce every local region that the sign of flow is clockwise, the normalization is performed by flipping the regions whose signs are counter clockwise. In other words, the solution for whether to flip a region prior to feature extraction is based on the sign of C. For robustness, Eqn. 10 can be further enhanced by assigning higher weights to vectors closer to region center as following

$$C = \sum_{(x,y)\in I} \sqrt{\frac{\partial I(x,y)}{\partial x}^2 + \frac{\partial I(x,y)}{\partial y}^2} \times \cos\theta \times G(x,y,\sigma) \tag{10}$$

where the flow is weighted by a Gaussian kernel G of size σ equal to the radius of local region[1].

To summarize, F-SIFT generates descriptors as following. Given a region rotated to its dominant orientation, Eqn. 10 is computed to estimate the flow direction of either clockwise or anti-clockwise. F-SIFT ensures flip invariance property by enforcing that the flows of all regions should follow a predefined direction indicated by the sign of C in Eqn. 10. For regions whose flows are opposite of the predefined direction, flipping the regions along the horizontal (or vertical) axis as well as complementing their dominant orientations are explicitly performed to geometrically normalize the regions. SIFT descriptors are then extracted from the normalized regions. In other words, F-SIFT operates directly on SIFT and preserves its original property. Selective flipping based on dominant curl analysis is performed prior to extracting flip invariant descriptor. Compared to SIFT, the overhead involved in F-SIFT is merely the computation of Eqn. 10 which is cheap to calculate. Our experimental simulation shows that the extraction of F-SIFT descriptors from an image is approximately one third slower than SIFT (See more details in Section IV-D). Figure 4 contrasts the matching performance of SIFT and F-SIFT for images undergone various transformations. The keypoints are extracted with Harris-Laplacian[2] detector and

[1] Following the convention of SIFT-like feature, local region is normalized to 41 × 41 and thus $\sigma = 20$.

[2] Code is available at http://www.cs.cityu.edu.hk/~wzhao2/lip-vireo.htm.

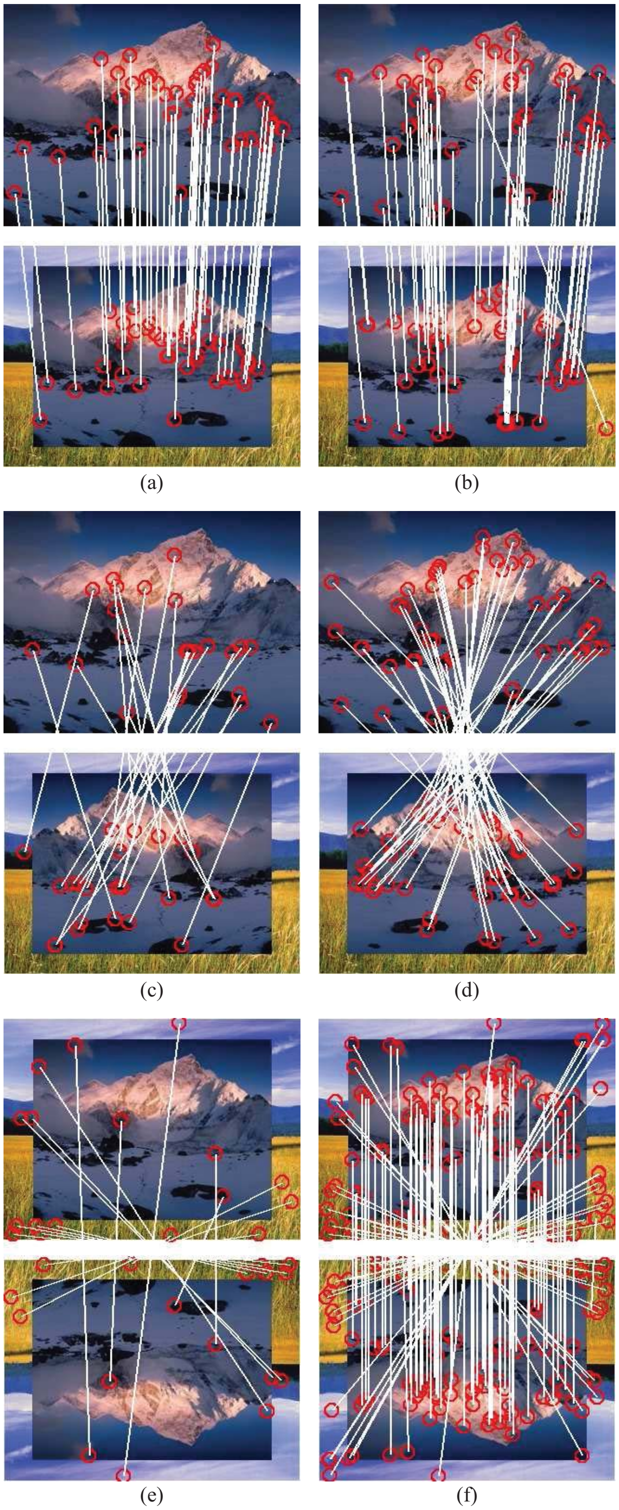


Fig. 4. Comparing the matching performance of SIFT (left) and F-SIFT (right) under flip transformations. (X/Y) shows the number of match pairs (X) against the number of keypoints (Y). For illustration purposes, not all matching lines are shown. (a) Scale (181/484). (b) Scale (162/484). (c) Scale+flip (24/484). (d) Scale+flip (153/484). (e) Flip+rotate (71/508). (f) Flip+rotate (307/508).

described with SIFT and F-SIFT respectively. The correspondences between points are matched through one-to-one symmetric matching algorithm (OOS) [22]. As shown in Figures 4(a) and 4(b), for transformation involving no flip, F-SIFT shows similar performance as SIFT. Fewer matching pairs are found however by F-SIFT as shown in 4(b) due to estimation error in Eqn. 10. The error comes from regions lacking of texture pattern. Conversely, when flip happens, F-SIFT exhibits significantly stronger performance than SIFT. As shown in Figures 4(c)-4(f), the number of matching pairs recovered by F-SIFT is much more than SIFT.

IV. Video Copy Detection

To demonstrate the use of F-SIFT for copy detection, we adopt our framework originally developed for near-duplicate video detection [23]. Modifications to the framework are made considering the new features introduced by F-SIFT. Following [23], F-SIFT descriptors are first offline quantized for generating visual vocabulary. Each keyframe extracted from videos is then represented as a bag-of-visual-words (BoW) indexed with inverted file structure (IF) for fast online retrieval. For reducing quantization loss, each word indexed by IF is also associated with Hamming signature for robust filtering [15], [24]. In addition, geometric checking is employed to prune falsely retrieved keyframes [23]. Finally, the detected keyframes of a candidate video are aggregated and aligned with query videos by Hough Transform [15], [23].

A. Indexing F-SIFT

An interesting fact, when matching a flipped image with its original copy, is that the flow directions of two matched regions computed by Eqn. 10 are always opposite of each other. Recall that F-SIFT makes the extracted descriptors flip invariant by explicitly flipping one of the regions before feature extraction. Conversely, when the transformation on images does not involve flip, the matched regions are either not flipped or both flipped by F-SIFT. In other words, in ideal cases, there are only two possibilities to describe the matches between two images. First, in the case when a query image is flipped, the matched pairs all have the characteristics that one of the regions is flipped by F-SIFT. Second, when a query involves no flip, all the matched pairs are either not flipped or flipped but not a mixture of them. While this observation is intuitive, it leads to the interesting idea that false matches can be easily pruned. For example, by surveying all the matched pairs from two images and finding out which of the two possible cases (query is flipped or not flipped), invalid matches can be easily identified and removed.

We make use of this simple fact to revise the inverted file (IF) structure by also indexing whether a salient region is flipped by F-SIFT. In addition to the spatial location, scale, orientation and Hamming signature [24] of a keypoint to be indexed by IF, an extra bit, of value equals to either 1 or 0 for indicating flip or otherwise, is required. During online retrieval, the retrieved visual words, together with their flip indicators, are consolidated for finding out which of two possible cases that the majority matches belong to. The remaining matches could then be treated as invalid matches and removed from further processing. For example, in the case when a query is regarded as not involving flip operation, all matched words with different bit values are directly pruned. While simple, this strategy easily filters significant amount of false positive matches and speeds up the online retrieval as demonstrated in our experiments (see Section IV-D). In addition, since only one bit is required for flip indicator, the space overhead to IF is kept in minimal. Note that the use of flip indicator is analog to the use of Laplacian sign in [20], except the former is for verifying matches while the latter is mainly for speeding up matching.

B. Enhanced Weak Geometric Consistency Checking

The retrieved visual words by IF could still be noisy in general due to quantization error. A practical approach for reducing noise is by weakly recovering the underlying geometric transformation [23]–[25] for further verification. We adopt E-WGC in [23] for geometric checking due to its superior performance compared to the more established approach WGC [24]. With the use of F-SIFT, we revise the E-WGC as following. Given two matched visual words $q(x_q, y_q)$ and $p(x_p, y_p)$ from a query and a reference keyframe respectively, the linear transformation between them can be expressed as

$$\begin{bmatrix} x_q \\ y_q \end{bmatrix} = s \times \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}. \tag{11}$$

There are three parameters to be estimated in Eqn. 11: the scaling factor s, the rotation parameter θ, and the translation T_x, T_y. In E-WGC, Eqn. 11 is manipulated as

$$\begin{bmatrix} \widetilde{x_q} \\ \widetilde{y_q} \end{bmatrix} = \widetilde{s} \times \begin{bmatrix} \cos\widetilde{\theta} & -\sin\widetilde{\theta} \\ \sin\widetilde{\theta} & \cos\widetilde{\theta} \end{bmatrix} \times \begin{bmatrix} x_p \\ y_p \end{bmatrix} \tag{12}$$

where $\widetilde{s} = 2^{s_q - s_p}$ and $\widetilde{\theta} = \theta_q - \theta_p$. The notations s_q and θ_q represent the characteristic scale and dominant orientation of visual word q respectively.

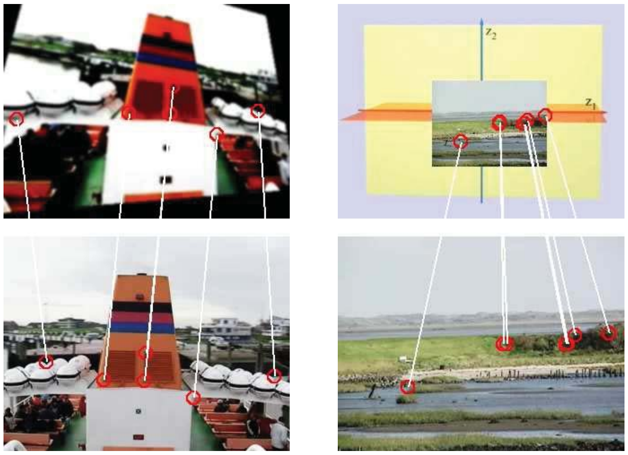
When a query is regarded as a flip version of a reference image in database as discussed in Section IV-A, Eqn. 12 is rewritten as

$$\begin{bmatrix} \widetilde{x_q} \\ \widetilde{y_q} \end{bmatrix} = \widetilde{s} \times \begin{bmatrix} \cos\widetilde{\theta} & -\sin\widetilde{\theta} \\ \sin\widetilde{\theta} & \cos\widetilde{\theta} \end{bmatrix} \times \begin{bmatrix} W_0 - x_p \\ y_p \end{bmatrix} \tag{13}$$

where W_0 is the width of reference image. Note that Eqn. 13 considers only horizontal reflection[3] for speed efficiency. The choice of applying either Eqn. 12 or Eqn. 13 is determined on the fly based on whether flip transformation is detected as presented in Section IV-A. E-WGC aims to estimate the translation τ of visual word q by

$$\tau = \sqrt{(\widetilde{x_q} - x_q)^2 + (\widetilde{y_q} - y_q)^2} \tag{14}$$

which can be efficiently estimated by histograming technique. Specifically, the value of τ computed from any two matched visual words are hashed to a histogram. The peak of histogram reflects the dominant translation between two images, indicating that any matches that do not fall into the peak will eventually be treated as false positives and pruned.



(a) (b)

Fig. 5. Examples of copies with very few true positive matches due to heavy transformation. (a) Heavy skew. (b) Large scaling.

C. Reciprocal Geometry Verification

Given the valid visual word matches returned by IF and E-WGC verification, the similarity between a query Q and a reference image R is given by

$$Sim(Q, R) = \frac{\sum h(q, p)}{\|BoW(Q)\|_2 \cdot \|BoW(R)\|_2} \tag{15}$$

where $h(q, p)$ is the distance [24] between Hamming signatures of q and p. The notation $BoW(Q)$ denotes the bag-of-words of Q. Notice that, because of the aggregated Hamming distances, the value given by Eqn. 15 can exceed *1*. In order to evaluate the similarity between query and reference video, similarities of matched query and reference keyframes are aggregated on $Sim(Q, R)$ via Hough transform [15], [23].

In practice, Eqn. 15 is not robust to heavy transformation which often causes few matches between two keyframes. Figure 5 shows an example where there are only six matches being identified due to large skew and scale resulting in low similarity scores by Eqn. 15. To alleviate this problem, we revise $h(p, q)$ in Eqn. 15 such that the similarity is not only dependent on Hamming distance but also the confidence of matching between two words. In this way, keyframe pairs with few matches could also be ranked high in the resulting list. The $h(p, q)$ is revised as

$$H(q, p) = (1.0 - \Delta) \times \log_\alpha \Delta \times h(q, p) \tag{16}$$

where Δ indicates the confidence of matching which will be further elaborated later, and α is an empirical parameter which is set to 0.9 in our experiment. Eqn 16 basically amplifies $h(q, p)$ when the matched pair holds high confidence score (low Δ in another word).

We estimate Δ by reciprocal geometric verification. Given two matched words p and q from keyframes Q and R respectively, the scale $\hat{s}$ and rotation $\hat{\theta}$ between them can be approximated by referring to another matched words of p'

[3] Horizontal flip is more commonly observed than reflection along other directions. This is due to the fact that mirror-like transformation visually will not result in apparent loss of visual content. Furthermore, scenes capturing from two opposite viewpoints, which happen frequently in news videos, also simulate mirror effect.

TABLE I

COMPARISON OF F-SIFT AND SIFT FOR VIDEO COPY DETECTION UNDER DIFFERENT TYPES OF TRANSFORMATIONS: 1) CAMCODING, 2) PICTURE-IN-PICTURE, 3) INSERTION OF PATTERNS, 4) STRONG RE-ENCODING, 5) CHANGE OF GAMMA, 6) DECREASE IN QUALITY (INCLUDING NOISE, FRAME DROPPING, ETC.), 8) POST PRODUCTION (INCLUDING INSERTION OF CAPTIONS, FLIPPING, ETC.), 10) RANDOMLY CHOOSE ONE TYPE FROM 3 MAJOR TRANSFORMATIONS. THE 3rd COLUMN INDICATES THE NUMBER OF COPY VIDEOS CORRECTLY RETRIEVED UNDER DIFFERENT TRANSFORMATIONS. NOTE THAT IN TRANSFORMATION 8 AND 10, THERE ARE 63 AND 14 OUT OF 134 QUERIES BEING FLIPPED, RESPECTIVELY

(a) COMPARISON AMONG DIFFERENT SETTINGS OF BOW

Transformations	Options	1	2	3	4	5	6	8	10	Prec	Rec
BoW* [24]	SIFT	6	8	79	4	73	14	44	23	0.234	0.218
	F-SIFT	4	6	87	6	84	16	69	33	0.285	0.285
BoW	SIFT	9	60	112	34	110	45	61	51	0.450	0.457
	F-SIFT	13	77	120	55	122	54	114	68	0.581	0.548
BoW+	SIFT	49	79	124	67	122	72	77	66	0.612	0.609
	F-SIFT	**54**	**91**	**126**	**78**	**127**	**85**	**117**	**81**	**0.708**	**0.719**

(b) COMPARISON BETWEEN SIGN OF DOMINANT CURL AND SIGN OF LAPLACIAN USING F-SIFT

Transformations	1	2	3	4	5	6	8	10	Prec	Rec
BoW* [24]	4	6	87	6	84	16	69	33	0.285	0.285
+Dominant curl	1	19	98	1	80	24	71	35	0.635	0.307
+Laplacian	0	11	92	0	75	21	67	29	0.596	0.275

from Q and q' from R, where

$$\hat{s} = \frac{|\overrightarrow{pp'}|}{|\overrightarrow{qq'}|} \tag{17}$$

$$\hat{\theta} = \overrightarrow{qq'}\angle\overrightarrow{pp'}. \tag{18}$$

Notice that $\hat{s}$ and $\hat{\theta}$ could be different from the values $\widetilde{\theta}$ and $\widetilde{s}$ estimated by keypoint detection (as given in Eqn. 12). However, in general the closer their values are, the higher chance that the match between p and q is correct. We thus define Δ as the discrepancy value between them as $\Delta = max\{|\hat{\theta} - \widetilde{\theta}|, |\hat{s} - \widetilde{s}|\}$. Basically the smaller the value is, the more confidence is for the match between words p and q. For any value where $\Delta \geq \alpha$, the match will be directly removed from similarity measure such that Eqn. 16 will always produce positive value. Referring back to equations Eqn. 15 and Eqn. 16, the similarity between two keyframes is revised by weighting the significance of matched words based on their Hamming distance and matching confidence.

D. Experiment

The experiments are conducted on TRECVID [9] sound and vision dataset 2010. The dataset consists of *11,525* web videos with a total duration of *400* hours. There are *1,608* queries which are artificially generated by eight different transformations ranging from camcording, picture-in-picture, re-encoding, frame dropping to the mixture of different transformations including flip. For pre-processing, dense keyframe sampling is performed on both query and reference videos with the rate of one keyframe per *1.6* seconds. This results in an average of *51* keyframes per query, and a total of *903,656* keyframes in the reference dataset. We employ Harris-Laplacian for keypoint detection and there are *309* keypoints per frame on average. For BoW representation, we adopt binary quantization and multiple assignment of visual words to a keypoint [6]. Comparing to hard quantization, binary quantization exhibits much better robustness towards the phenomenon of burstiness [26] which widely exists across images and video frames. For page limitation, the results from hard quantization are omitted. For each query, a copy video is returned (if there is any) with a similarity score.

The evaluation follows the way TRECVID CCD takes. For each type of transformation, a recall-precision curve is generated. An optimal threshold is selected at the point these two measures are balanced. The performance is evaluated based on recall and precision at this optimal truncation point.

Detection Effectiveness: We compare the performance of F-SIFT and SIFT under three different settings: BoW+, BoW and BoW*, in order to see the effect of different components on the visual descriptors. BoW+ is the proposed framework in this paper, while BoW includes all the features discussed in this section except reciprocal geometric verification. BoW* is implemented based on [6], [24] which reported excellent performance on TRECVID datasets by visual word matching and is widely regarded as the state-of-the-art technique on video copy detection. BoW* basically represents a more conventional framework in the literature where, different from the BoW setting, the sign of dominant curl is not utilized for filtering and geometric verification is based on WGC.

Table I(a) shows the performance comparison for *1,608* queries over eight different transformations. Based on the ground-truth provided by CCD, there are *134* copies per transformation. As indicated by the results, F-SIFT outperforms SIFT by consistently returning more true positives almost across all types of transformations and settings. Especially for transformation-8 and transformation-10 which involve flip operation, F-SIFT detects 62.5%, 38.5% and 52.3% more true positives under BoW, BoW+ and BoW$*$ respectively. It is worth notice that while F-SIFT is built upon SIFT, it is also capable of exhibiting similar or even better performance for no-flip transformations. Comparing all three

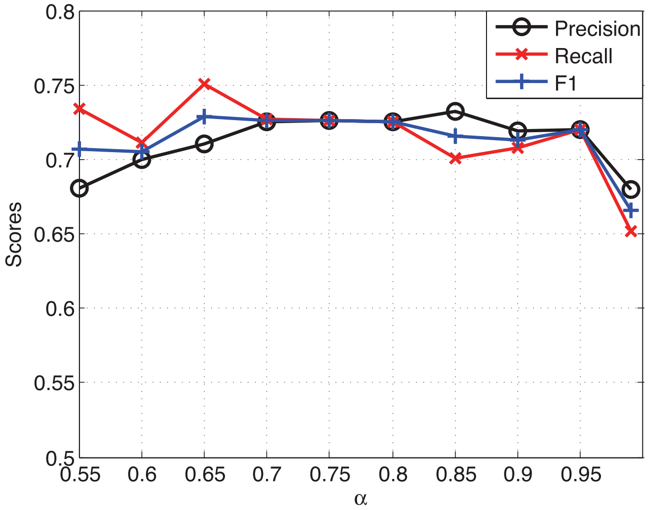
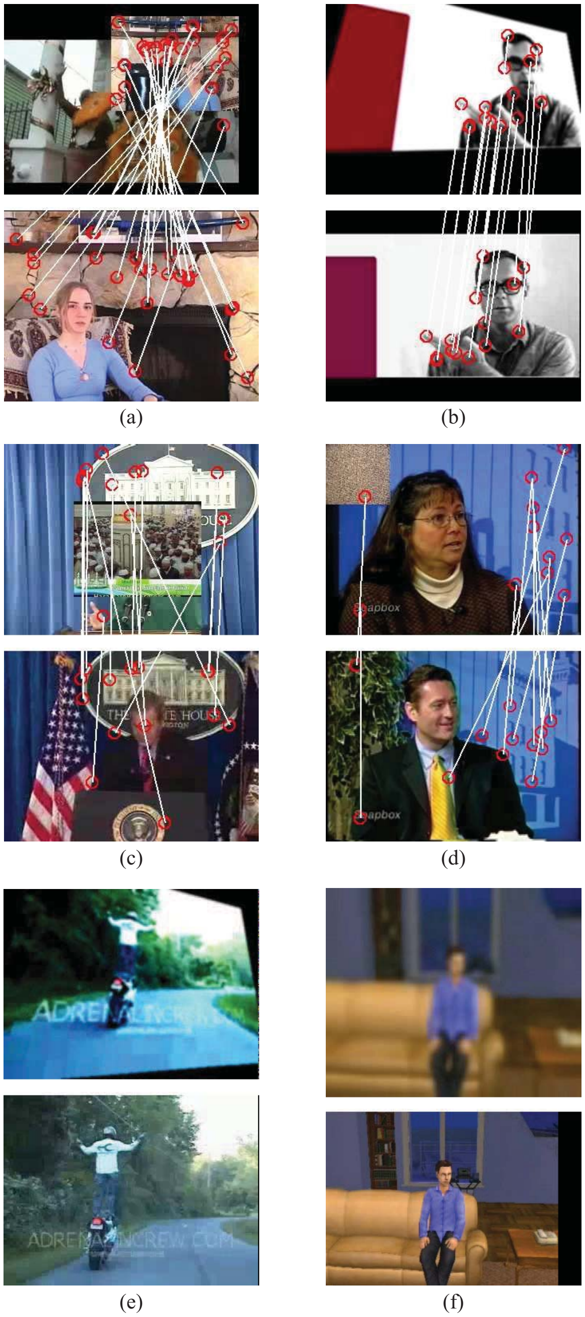


Fig. 6. Sensitivity of α in (16) towards the performance of video copy detection.

settings, the performance of BoW is better than BoW∗ due to the use of E-WGC instead of WGC. The use of flip indicators in BoW for pruning false matches also leads to larger degree of improvement in detection precision. Further incorporating reciprocal geometric checking as by BoW+ leads to the overall best performance in the experiment. By examining the similarity scores between queries and true positives, we confirm that the use of re-weighting strategy by Eqn. 16 has successfully boosted the ranking of candidate videos with fewer true matches. Note that Eqn. 16 involves a parameter α which is empirically set to *0.9* in the experiment. Figure 6 shows the sensitivity of α, where as long as the value falls within the range of $[0.65, 0.95]$, α is not sensitive to the performance.

The idea of adopting sign of dominant curl for fast filtering of false alarms is analog to the use of sign of Laplacian for fast matching of visual words in [20]. Table I(b) shows the performance comparison between them based on BoW* setting. Overall, performance improvement in both recall and precision is observed when enhancing BoW* with sign of dominant curl. This is in contrast to using the sign of Laplacian which improves precision of BoW* but degrades recall. Because the purpose of using sign of Laplacian is mainly for speeding up [20], it is less effective in keeping correct matches and less capable of dealing with flip transformation compared to dominant curl. Over the eight transformations, sign of dominant curl consistently exhibits better ability in recalling true positives.

Figure 7 shows the examples of match results produced by F-SIFT under BoW+ setting. In general, F-SIFT is robust to scaling, flipping and skew transformations as shown in Figure 7(a) and 7(b). By manual checking, most of matches are correct. False positives, as shown in Figures 7(c) and 7(d), are generated however due to partial scene duplicate. While the results are regarded as false alarms, by manual checking we can find that the duplicate object and background are indeed correctly matched. False negatives are produced mainly due to blur transformation. There are barely no keypoint matches found using either F-SIFT or SIFT for the examples in Figures 7(e) and 7(f). We investigate the result and observe that this is mainly due to quantization error introduced by BoW quantization[4].


(a) (b)

(c) (d)

(e) (f)

Fig. 7. Example of matching results by F-SIFT. (a) and (b) True positives. (c) and (d) False positives. (e) and (f) False negatives. (a) Flip+scale. (b) Skew. (c) Duplicate object. (d) Duplicate background. (e) Heavy skew+blur. (f) Heavy blur+scale.

Efficiency: Table II lists the time cost for processing one query keyframe. The experiments are conducted on a PC with *2.8*GHz CPU and *7*G main memory under Linux environment. In terms of feature extraction time, F-SIFT takes additional *0.128* seconds compared to SIFT. During the retrieval stage

[4] When matching the F-SIFT features directly (instead of using BoW) using one-to-one symmetric matching [22], there are plenty of correct matches being found, for examples in Figures 7(e) and 7(f).

TABLE II
AVERAGE TIME COSTS IN EACH STEP OF CCD FOR SIFT AND F-SIFT BASED APPROACHES (S)

	SIFT	F-SIFT
Feature Extraction	0.651	0.779
Binary VQ	0.209	0.209
Retrieval by IF	0.162	0.218
E-WGC	1.412	0.553
Reciprocal Verification	0.311	0.167
Total	2.802	1.938

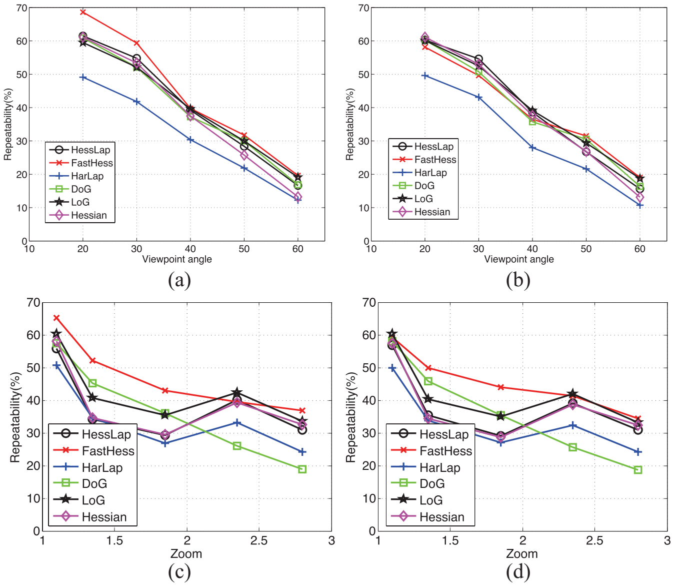
by inverted file (IF), F-SIFT is also slower than SIFT due to the need for consolidating the matching result by checking the flip indicators of matched words. However, this step effectively prunes false matches and results in much less candidates to be further processed by E-WGC and reciprocal geometric verification. As shown in Table II, the computation time is reduced by *61%* for E-WGC, and by *46%* for reciprocal geometry verification. This ends up with a more efficient and effective video copy detection framework using F-SIFT. In our dataset of *0.9* million keyframes, processing a typical query of *71.6* seconds with *51* keyframes will take about *98.8* seconds by F-SIFT. This speed up is *44.6%* comparing to SIFT which will take *142.9* seconds.

V. OBJECT RECOGNITION

The effectiveness of local features towards recognizing objects under different degree of transformations has been surveyed in [11]. In this section, we conduct similar studies to compare the recognition effectiveness of different keypoint detectors and descriptors. Particularly, we investigate the performance of F-SIFT in comparison to various visual descriptors in dealing with flip and no-flip transformations. The following experiments are conducted based on the image sequences and testing software provided by K. Mikolajczyk [11], [18].

Keypoint Detector: The aim of this experiment is to empirically study the flip invariance property of keypoint detectors as presented in Section III-A. Two image sequences, *Wall* and *Boat*, as well as their flip versions are used for experiment. The former is a sequence showing the gradual change of viewpoint, while the latter shows the gradual change of zoom and rotation. We evaluate six different keypoint detectors and compare their performances based on repeatability rate [18]. Figure 8 shows the results. As noted, the performance trends for all the six detectors are consistently similar in both flip and no-flip transformation. The empirical result therefore coins with the analysis in Section III-A that most of the existing detectors are flip invariant.

Visual Descriptor: We compare eight different visual descriptors including F-SIFT and SIFT for investigating their accuracy in keypoint matching. Similar to [20], a set of image pairs are sampled from the eight image sequences for experiment. The set includes the first and fourth images from each sequence. In addition to the original transformations (blur, rotation, zoom, change of lighting, color and JPEG compression rate) in the image set, flip transformation is included by flipping the fourth image of each sequence. In the experiment, except SURF, DoG detector is employed for all the visual descriptors. Following [11], the performance evaluation is measured by assessing the number of point-to-point matches being correctly returned. Figure 9 shows the performance in terms of recall-precision curve averaged over the results on eight image pairs. As shown in Figure 9(a), for no-flip transformation, SIFT exhibits the best performance followed by F-SIFT and SURF. In the worst case, F-SIFT is still able to achieve 85% performance of SIFT, which is far better than other descriptors such as MI-SIFT and FIND designed for dealing with flips. The performance degradation of F-SIFT is mainly due to the errors in dominant curl estimation. For flip transformation as shown in Figure 9(b), conversely, F-SIFT shows superior performance than the popular descriptors such as SIFT, SURF and PCA-SIFT which, as F-SIFT, also use directionally sensitive gradient feature. Although RIFT and SPIN share similar partitioning scheme (see Figure 2(b) and 2(d)), SPIN demonstrates much stable performance for preserving flip invariance property. However, SPIN suffers from low visual distinctiveness due to the use of pixel intensity rather than directional gradient as feature. As a result, the performance is not as good as F-SIFT. Similarly for MI-SIFT, which uses moment as feature, the performance is also not satisfactory and lower than it was originally shown in [12]. FIND, on the other hand, exhibits relatively stable performance in both flip and no-flip cases. Nevertheless, its partitioning scheme (see Figure 2(e)) appears to be less effective than the conventional grid-based partitioning such as adopted by SIFT and SURF, which has limited its overall performance.

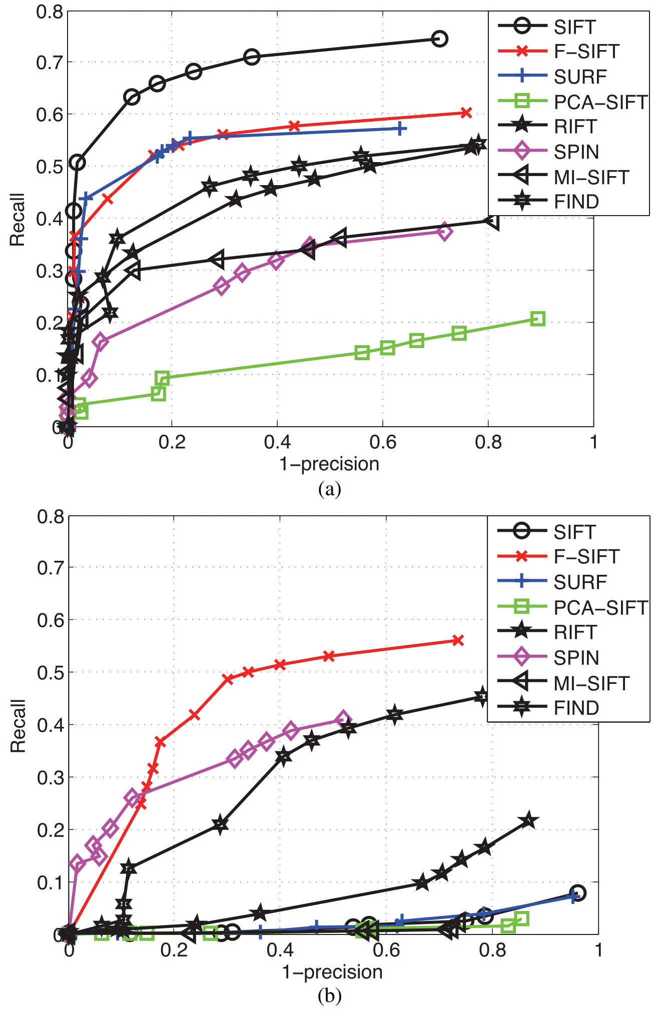


(a) (b) (c) (d)

Fig. 8. Repeatability of various keypoint detectors on the original *Wall* and *Boat* sequences, and their flip versions in (b) and (d). (a) Wall. (b) Wall with flip. (c) Boat. (d) Boat with flip.

VI. OBJECT DETECTION

Visual object detection has been extensively studied in recent ten years. Among variants of approaches, detection



(a)

(b)

Fig. 9. Comparison of eight different visual descriptors for flip and no-flip cases using the image pairs sample from eight image sequences [11]. (a) No-flip. (b) Flip.

based on bag-of-words representation and SVM classifiers has been the most popularly adopted technique. In this section, we experimentally compare F-SIFT and SIFT for this detection paradigm. Particularly, we adopt a variant of BoW with Fisher kernel as framework, which has been shown to generate the state-of-the-art classification performance on large-scale image dataset in [2], [3].

A. Fisher Kernel on BoW

Based on [2], each visual word is modeled as a GMM (Gaussian mixture model). The set of keypoint descriptors (e.g., F-SIFT), denoted as X, extracted from an image can be characterized by the following gradient vector:

$$\bigtriangledown \log p(X|\lambda) \tag{19}$$

where $X = [x_1, x_2, \ldots, x_t]$ has t keypoints and λ is the parameter set characterizing GMM. Intuitively, the gradient of the log-likelihood describes the direction in which parameters should be modified to best fit the data. The attractiveness of Fisher kernel is the transformation from a variable length sample X into a fixed length vector determined by λ. The gradient vector can then be treated as input for any type of classifier. Typically, this gradient vector can be divided into three sub-vectors by its parameter types: weight (w_i), mean (μ_i) and variance (σ_i) ($i = 1, \ldots, N$), where N is the number of Gaussians or visual words. For instance, the gradient on μ_i can be approximated by

$$\frac{\partial L(X|\lambda)}{\partial \mu_i^d} = \sum_{s=1}^{t} [x_s^d - \mu_i^d] \tag{20}$$

where d is the dimension of descriptor. In our implementation, only mean gradient sub-vector is employed for classification since its performance is similar to that of using all three sub-vectors [2]. The advantages of Fisher kernel based BoW are twofold. First, smaller number of visual words are required compared to BoW. Second, since the feature space has been unfolded, more efficient classifiers such as linear SVM can be employed, which was demonstrated in [2], [3] to achieve similar performance as nonlinear classifier. Eqn. 20 has also been successfully employed in large-scale content-based image retrieval [27].

B. Experiment

We conduct experiments on PASCAL VOC 2009 dataset [28]. There are *20* object classes and *23,074* images crawled from Flickr. The dataset is split into two parts: *7,054* images for training and *6,650* images for testing. The performance evaluation is measured by average precision (AP).

We compare the performance of F-SIFT and SIFT based on keypoints extracted from five different detectors: Harris-Laplace (HarLap), Hessian-Laplace (HessLap), Hessian, Difference-of-Gaussian (DoG) and Laplacian-of-Gaussian (LoG). For Fisher kernel based BoW, a small visual vocabulary of *80* words is generated and linear SVM is employed for classification. Table III lists the performance of object detection. As shown in Table III, F-SIFT outperforms SIFT for most of the object classes, and more importantly, the improvement is consistently observed across all the five keypoint detectors. Among them, F-SIFT descriptors extracted from Harris-Laplace and Hessian-Laplace detected keypoints achieve the highest mean AP. The improvement introduced by F-SIFT indicates the existence of symmetric structures in object classes which are well described by F-SIFT than SIFT. Nevertheless, performance drop is also observed in few classes. From our analysis, the performance fluctuation between F-SIFT and SIFT has no correlation to any particular object classes. The performance improvement or degradation by F-SIFT is more closely related to the type of keypoint detector being employed. For instance, DoG detector introduces less percentage of improvement than others, and there are *6* out *20* classes exhibit lower AP than SIFT. The performance drop is mainly because of the lack of texture pattern in some of keypoints detected by DoG. The computation of dominant curl is found to be less reliable for these keypoints, which affects the stability of F-SIFT. As a reference, we also compare the performance to more conventional implementation using BoW and SVM with RBF kernel. The results in terms of mean

TABLE III

Performance of F-SIFT and SIFT for Object Detection Using Fisher Kernel BoW and Linear SVM. The Comparison Is Made Against Different Types of Keypoint Detectors. The Items in Parentheses in the Last Row Indicate the Mean AP by Using Standard BoW and SVM With RBF Kernel

Detector	HarrLap		HessLap		Hessian		DoG		LoG	
Class	SIFT	F-SIFT	SIFT	F-SIFT	SIFT	F-SIFT	SIFT	F-SIFT	SIFT	F-SIFT
Aeroplane	0.737	**0.747**	0.732	0.721	0.696	0.719	0.672	0.696	0.707	0.732
Bicycle	0.308	0.314	0.427	**0.441**	0.398	0.403	0.386	0.369	0.405	0.427
Bird	0.377	**0.392**	0.325	0.342	0.298	0.283	0.332	0.358	0.34	0.325
Boat	0.455	**0.499**	0.3335	0.3532	0.3188	0.3342	0.315	0.323	0.3391	0.334
Bottle	**0.226**	0.2	0.216	0.213	0.206	0.179	0.133	0.128	0.219	0.216
Bus	0.496	0.516	0.511	0.525	0.488	**0.549**	0.463	0.505	0.492	0.511
Car	0.301	0.336	0.4	0.41	0.376	**0.407**	0.357	0.361	0.367	0.4
Cat	**0.451**	0.446	0.413	0.425	0.379	0.402	0.336	0.345	0.393	0.413
Chair	0.37	**0.389**	0.37	0.364	0.346	0.354	0.347	0.356	0.348	0.37
Cow	**0.233**	0.214	0.187	0.151	0.137	0.16	0.126	0.155	0.18	0.187
Diningtable	0.181	0.237	0.243	**0.267**	0.231	0.208	0.198	0.226	0.25	0.243
Dog	0.31	**0.322**	0.305	0.335	0.284	0.291	0.298	0.299	0.304	0.305
Horse	0.298	0.321	0.302	**0.342**	0.29	0.327	0.324	0.344	0.269	0.302
Motorbike	0.342	0.352	0.457	**0.488**	0.448	0.446	0.412	0.439	0.433	0.457
Person	0.701	**0.716**	0.684	0.702	0.67	0.678	0.659	0.671	0.681	0.684
Pottedplant	0.12	0.114	0.138	0.111	0.119	0.12	0.159	**0.166**	0.09	0.138
Sheep	**0.257**	0.251	0.221	0.196	0.188	0.193	0.127	0.1	0.225	0.221
Sofa	0.227	**0.25**	0.212	0.247	0.16	0.225	0.216	0.215	0.198	0.212
Train	0.507	0.521	0.509	0.509	0.453	0.538	**0.551**	0.54	0.459	0.509
TVmonitor	0.332	0.351	**0.394**	0.343	0.366	0.377	0.355	0.344	0.393	**0.394**
Mean AP	0.362	**0.374**	0.369	**0.374**	0.343	0.360	0.338	0.347	0.355	0.369
	(0.371)	(0.385)	(0.347)	(0.363)	(0.342)	(0.354)	(0.353)	(0.358)	(0.335)	(0.345)

Bold font indicates the best performance for each class.

AP are shown in the last row of Table III. Basically, using F-SIFT also leads to similar performance gain, and we observe no significant difference in AP performance between these two implementations. This also indicates that performance of F-SIFT is stable over different versions of BoW and SVM.

VII. Conclusion

We have presented F-SIFT and its utilization for video copy detection, object recognition and image classification. On one hand, the extraction of F-SIFT is slower than SIFT due to the computation of dominant curl and explicit flipping of local region. On the other hand, the improvement in detection effectiveness is consistently observed in three applications. Video copy detection, in particular, demonstrates significant improvement in recall and precision with the use of F-SIFT. More importantly, by wisely indexing the F-SIFT with extra overhead of one bit per descriptor in space complexity, the speed of online detection (excluding feature extraction) on a dataset of *0.9* million keyframes has also been improved by about two times. This indeed has compensated the need for longer time in feature extraction.

In copy detection, we demonstrate the use of F-SIFT in predicting whether a query is a flipped version of a reference video. As shown in our experiments, this interesting finding has led to significant speed up by reducing large amount of candidate matches for post-processing. In object recognition, the comparative study shows that F-SIFT outperforms seven other visual descriptors when flip is introduced on top of various transformations, while exhibiting similar performance as SURF for no-flip transformation. In object detection, it is also possible to take advantage of F-SIFT for analyzing the flip-like structure in image and improving detection effectiveness. Our future work thus includes the exploitation of F-SIFT for more comprehensive and explicit way of describing symmetric patterns latent in objects.

Acknowledgment

The authors would like to thank Dr. R. Ma from IBM, Beijing, China, for kindly sharing the mirror and invert invariant scale-invariant feature transform source code. They would also like to thank Mr. X.-J. Guo from Tianjin University, Tianjin, China, whose informative suggestions and patience helped with the implementation of the FIND descriptor.

References

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[2] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[3] F. Perronnin, J. Sańchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.

[4] M.-C. Yeh and K.-T. Cheng, "A compact, effective descriptor for video copy detection," in *Proc. Int. Conf. Multimedia*, 2009, pp. 633–636.

[5] Z. Liu, T. Liu, D. Gibbon, and B. Shahraray, "Effective and scalable video copy detection," in *Proc. Int. Conf. Multimedia Inf. Retr.*, 2010, pp. 119–128.

[6] M. Douze, H. Jégou, and C. Schmid, "An image-based approach to video copy detection with spatio-temporal post-filtering," *IEEE Trans. Multimedia*, vol. 12, no. 4, pp. 257–266, Jun. 2010.

[7] Y.-D. Zhang, K. Gao, X. Wu, H. Xie, W. Zhang, and Z.-D. Mao, "TRECVID 2009 of MCG-ICT-CAS," in *Proc. NIST TREVCID Workshop*, 2009, pp. 1–11.

[8] J. Law-To, A. Joly, and N. Boujemaa. (2007). *Muscle-VCD-2007: A Live Benchmark for Video Copy Detection* [Online]. Available: http://www-rocq.inria.fr/imedia/civr-bench/

[9] *TRECVID*. (2008) [Online]. Available: http://www-nlpir.nist.gov/projects/trecvid/

[10] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1265–1278, Aug. 2005.

[11] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[12] R. Ma, J. Chen, and Z. Su, "MI-SIFT: Mirror and inversion invariant generalization for SIFT descriptor," in *Proc. Int. Conf. Image Video Retr.*, 2010, pp. 228–236.

[13] X. Guo and X. Cao, "FIND: A neat flip invariant descriptor," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 515–518.

[14] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in *Proc. Int. Conf. Multimedia Inf Retr.*, 2006, pp. 321–330.

[15] M. Douze, A. Gaidon, H. Jégou, M. Marszatke, and C. Schmid, "INRIA-LEAR's video copy detection system," in *Proc. NIST TREVCID Workshop*, 2008, pp. 1–8.

[16] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[17] J. Deng, A. C. Berg, K. Li, and F.-F. Li, "What does classifying more than 10 000 image categories tell us?" in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 71–84.

[18] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.

[19] T. Linderberg, "Feature detection with automatic scale selection," *Int. J. Comput. Vis.*, vol. 30, no. 2, pp. 79–116, 1998.

[20] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[21] G. Strang, *Caculus*. Cambridge, MA: Wellesley-Cambridge, 1991, pp. 589–590.

[22] W.-L. Zhao, C.-W. Ngo, H.-K. Tan, and X. Wu, "Near-duplicate keyframe identification with interest point matching and pattern learning," *IEEE. Trans. Multimedia*, vol. 9, no. 5, pp. 1037–1048, Aug. 2007.

[23] W.-L. Zhao, X. Wu, and C.-W. Ngo, "On the annotation of web videos by efficient near-duplicate search," *IEEE Trans. Multimedia*, vol. 12, no. 5, pp. 448–461, Aug. 2010.

[24] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 304–317.

[25] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[26] J. Hervé, D. Matthijs, and S. Cordelia, "On the burstiness of visual elements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1169–1176.

[27] H. Jégou, M. Douze, C. Schmid, and P. Pèrez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3304–3311.

[28] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (2009). *The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results* [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html

Wan-Lei Zhao received the B.Eng. and M.Eng. degrees from the Department of Computer Science and Engineering, Yunnan University, Kunming, China, in 2006 and 2002, respectively, and the Ph.D. degree from the City University of Hong Kong, Kowloon, Hong Kong, in 2010.

He was with the Software Institute, Chinese Academy of Science, Beijing, China, from 2003 to 2004, as an Exchange Student. He was with the University of Kaiserslautern, Kaiserslautern, Germany, in 2011. He is currently a Post-Doctoral Researcher with INRIA-Rennes, Rennes, France. His current research interests include multimedia information retrieval and video processing.

Chong-Wah Ngo (M'02) received the B.Sc. and M.Sc. degrees in computer engineering from Nanyang Technological University, Singapore, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong.

He was a Post-Doctoral Scholar with the Beckman Institute, University of Illinois at Urbana-Champaign, Champaign. He was a Visiting Researcher with Microsoft Research Asia. He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. His current research interests include large-scale multimedia information retrieval, video computing, and multimedia mining.

Dr. Ngo is currently an Associate Editor of the IEEE Transactions on Multimedia. He is the Program Co-Chair of the ACM Multimedia Modeling Conference 2012 and the ACM International Conference on Multimedia Retrieval 2012, and the Area Chair of the ACM Multimedia 2012. He was the Chairman of the ACM (Hong Kong Chapter) from 2008 to 2009.