Match-time covariance for descriptors

Eric Christiansen[1]
echristiansen@cs.ucsd.edu

Vincent Rabaud[2]
vrabaud@aldebaran-robotics.com

Andrew Ziegler[3]
amz@gatech.edu

David Kriegman[1]
kriegman@cs.ucsd.edu

Serge Belongie[1]
sjb@cs.ucsd.edu

[1] University of California, San Diego
La Jolla, California, USA

[2] Aldebaran Robotics
Paris, France

[3] Georgia Institute of Technology
Atlanta, Georgia, USA

Abstract

Local descriptor methods are widely used in computer vision to compare local regions of images. These descriptors are often extracted relative to an estimated scale and rotation to provide invariance up to similarity transformations. The estimation of rotation and scale in local neighborhoods (also known as steering) is an imperfect process, however, and can produce errors downstream. In this paper, we propose an alternative to steering that we refer to as match-time covariance (MTC). MTC is a general strategy for descriptor design that simultaneously provides invariance in local neighborhood matches together with the associated aligning transformations. We also provide a general framework for endowing existing descriptors with similarity invariance through MTC. The framework, Similarity-MTC, is simple and dramatically improves accuracy. Finally, we propose NCC-S, a highly effective descriptor based on classic normalized cross-correlation, designed for fast execution in the Similarity-MTC framework. The surprising effectiveness of this very simple descriptor suggests that MTC offers fruitful research directions for image matching previously not accessible in the steering based paradigm.

1 Introduction

Local descriptor methods are a fundamental technology in computer vision. They enable local regions to be compared, despite changes in viewpoint and appearance, and are used in applications such as structure-from-motion, object detection and recognition, and image retrieval. In the last decade, the accepted solution to the viewpoint invariance problem has been extract-time covariance (ETC), also known as canonization or steering [17, 18]. SIFT [8], SURF [2], ORB [15], BRISK [6], and FREAK [1] all use ETC. In this paradigm, the algorithm tries to estimate a canonical rotation and scale for the descriptor at the time of extraction.

ETC has several problems. First, viewpoint estimation is unreliable. This has been previously noted [5], and is demonstrated in Figure 2. This is empirically true even for SIFT scale
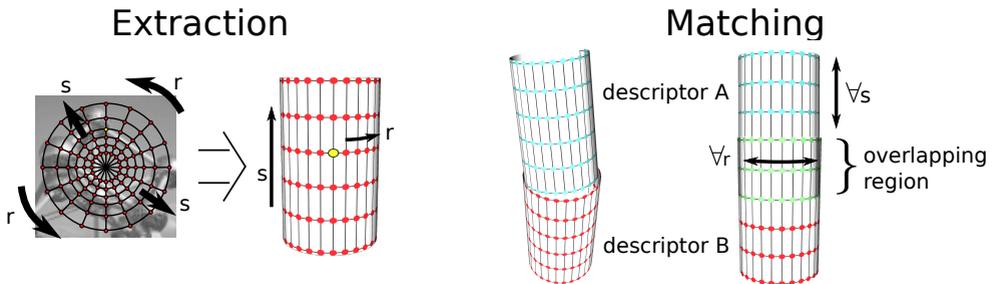
Figure 1: The Similarity-MTC extraction and matching framework, which provides any descriptor X with *invariance* and *offset*. This description is conceptual; actual implementations may be considerably more efficient. *In extraction*, descriptors are computed for a range of image scalings and rotations. To do this, a log-polar grid with N_S rings (scales) and N_R rays (rotations) is centered at each keypoint. For each grid intersection (s, r), the image is scaled and rotated about the keypoint to bring (s, r) to a canonical point, marked in yellow. Here (s, r) are integer grid coordinates, where $0 \le s < N_S$ and $0 \le r < N_R$. An X descriptor is then extracted at the keypoint location of the transformed image, and associated with the index (s, r), which is in turn associated with a point on a finite descriptor cylinder. *In matching*, two descriptor cylinders are aligned for all $(2N_S - 1)N_R$ possible ways. Optionally, a minimum overlap size N_O may be specified. For each alignment, a total distance is computed. The total distance is a function Ω of the descriptors in the overlapping regions; typically Ω is a normalized mean of the distances between corresponding descriptors. The final distance is the minimum total distance, and the similarity transformation is the relative cylinder motion that produced that distance. Note the brute-force cost of a full matching may be avoided, as in NCC-S. Best viewed in color.

estimation, despite theoretical work which suggests otherwise [10]. Second, because viewpoint estimation is unreliable, the relative pose induced by a corresponding keypoint pair cannot be trusted and is rarely used. Third, ETC requires a viewpoint estimation method, which adds runtime and code complexity. Fourth, a region's viewpoint cannot always be uniquely defined, as when there is circular symmetry about a detected point. These weaknesses of ETC have been overcome by increasing the number of analyzed descriptors (e.g. bag of features [16]), approximating the nearest neighbor search, or making the descriptors cheaper to compute (e.g. ORB, BRISK, and FREAK). Still, there are many cases that require few descriptors and/or accurate matching: structure from motion, tracking, or rigid object recognition with geometric constraints to name a few.

In this paper, we propose match-time covariance (MTC), an alternative to ETC. MTC is a general strategy for descriptor design, which simultaneously provides *invariance* and *offset*, the relative transformation to bring local regions into correspondence. The concept of MTC is not new; our contribution is to provide a novel synthesis of approaches related to MTC and to provide a general framework for endowing existing descriptors with similarity invariance through MTC. The framework, Similarity-MTC, is simple, dramatically improves accuracy, and additionally returns similarity transformations. Finally, we propose NCC-S, a remarkably simple descriptor that is designed for fast execution in the Similarity-MTC framework. The success of the descriptor also demonstrates the surprising effectiveness of plain normalized cross-correlation as a distance measure for wide-baseline matching.

The paper organization is as follows. Section 2 covers related works. Section 3 explains MTC, Similarity-MTC, and NCC-S. Section 4 demonstrates MTC's performance boost and provides timings. We conclude with Section 5.

2 Related Works

Match-time covariance (MTC) and the proposed NCC-S descriptor are simple ideas with a number of antecedents.

ASIFT is a wrapper around SIFT that endows SIFT with full affine invariance [11] by generating many synthetic warps of the local region, extracting a SIFT descriptor each time. The distance between two ASIFT descriptor bags B_1 and B_2 is $\min_{b_1 \in B_1, b_2 \in B_2} ||b_1 - b_2||_2$. ASIFT is an instantiation of MTC.

[20] and [13] estimate relative scale and rotation between images using log-polar coordinates. In this coordinate system, scalings and rotations correspond to translations. [20] cross-correlates log-polar transformed images, and recovers the relative scale and rotation from the coordinates of the maximum correlation. [13] cross-correlates log-polar sampled phase-only bispectrum slices, which additionally gives them blur invariance. However, they do not normalize by the descriptor overlap and so are not scale invariant. MTC also uses log-polar coordinates to obtain similarity invariance, and the NCC-S descriptor is simply a normalized version of [20].

The Scale Invariant Descriptor (SID) is a scale-*robust* descriptor that uses a log-polar sampling pattern and the Fourier transform modulus [5]. The log-polar pattern is used to sample feature responses obtained via band-pass filtering with the monogenic signal. These samples are then transformed into the Fourier domain, where phase information is discarded. This results in descriptors that can be directly compared using the l_2 distance. SID is very similar to NCC-S, but SID has partial invariance rather than match-time covariance, which reduces its sensitivity.

3 Match-time covariance with examples

We now define match-time covariance (MTC) and provide two examples: Similarity-MTC and NCC-S. Match-time covariance (MTC) is a simple strategy for descriptor design: when comparing two local regions (at matching time), we estimate the transformation that brings them into alignment (*offset*) and return a measure of dissimilarity between the aligned regions. In this paper, the transformation is assumed to be a geometric warp, but it could incorporate other transformations, such as blur. We call this *covariance* following the language of [17]. MTC differs from extract-time covariance (ETC) in that ETC tries to guess the alignment at extraction time. MTC addresses the unreliability of ETC ([5], Figure 2), but has wider implications, discussed in Section 5. To make MTC concrete, we provide two examples: Similarity-MTC, an instantiation of MTC for similarity transformations, and NCC-S, a descriptor designed for Similarity-MTC.

3.1 Similarity-MTC

Similarity-MTC is a general MTC framework for endowing existing descriptors with similarity invariance, and is described in Figure 1. Extraction is similar to ASIFT, warping the
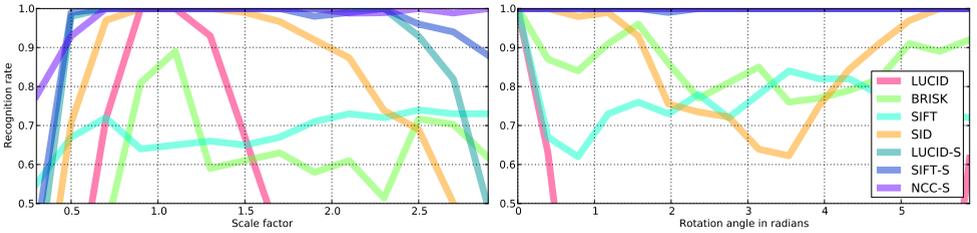
Figure 2: Recognition rates as a function of synthetic scale and rotation of the Oxford `boat` base image for various methods. The rates were obtained following the protocol of [22] and using the 100 strongest keypoints per image. The Similarity-MTC methods' high sensitivity and full similarity invariance give them the best performance. SID is not actually scale invariant; true scale invariance is elusive outside of MTC because of the need to normalize overlapping scale levels. SIFT and BRISK, as ETC methods, always pay a penalty for ETC unreliability. That penalty is most clearly captured in the difference between SIFT-S and SIFT. They differ only in covariance style; SIFT-S uses MTC and SIFT uses ETC. Thus, the performance gap is *exactly* the loss caused by ETC. All methods fail as image size goes to zero on the left side of the scale plot. For Similarity-MTC methods, the upper range of scale invariance is set by the parameter $R_{\max}$; see Section 3.2.1. Best viewed in color.

image for each transformation on a grid of transformations. In matching, Similarity-MTC compares all corresponding descriptors in the cylinder overlap region. This is in contrast to ASIFT, which considers only the minimum distance between any descriptor pair. By comparing all corresponding descriptors, Similarity-MTC avoids information loss that might result from suboptimal choice of descriptor scale and rotation. When the base descriptor distance is l_2, as for SIFT, SURF, and NCC-S, this comparison can be expressed as a cross-correlation. This enables a Fourier-space representation, and a corresponding complexity of $O(N_S N_R \log N_S N_R)$, where N_S and N_R are the number of scale levels and rotation gradations in the log-polar pattern. This is *faster* than the naive $O((N_S N_R)^2)$ ASIFT approach, which considers *less* information.

3.2 NCC-S

NCC-S is a straightforward mapping of NCC to Similarity-MTC. Its radical simplicity is inspired by BRIEF [5] and LUCID [22], contrasting with complex methods like SIFT.

NCC-S can be defined in terms of a dummy descriptor INTENSITY, which just extracts the intensity of a single pixel value at some fixed offset $(\varepsilon_x, \varepsilon_y)$ from the keypoint center. NCC-S is INTENSITY, wrapped in the Similarity-MTC framework, with Ω defined as the normalized correlation of the overlapping pixels. *The previous two sentences completely define NCC-S*, with complexity $O((N_S N_R)^2)$. We can reduce the complexity to $O(N_S N_R \log N_S N_R)$ by expressing the cylinder alignment step as cross-correlation (CC), allowing us to work in Fourier space. The CC is then normalized using an approach akin to [7]. The optimized extraction and matching steps are described in Figure 3, with additional details below.

In *optimized extraction*, three summary statistics are computed for each block X of the $2N_S - 1$ possible blocks that could overlap in the cylinder alignment step. The statistics are: the average value $\langle X_{ij} \rangle$, the centered Frobenius norm $||X - \langle X_{ij} \rangle J||_F$, and $\sum \tilde{X}_{ij}$. Here J is
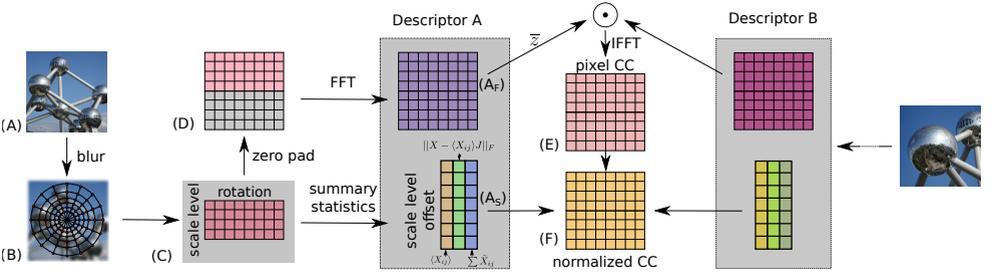
Figure 3: Fast NCC-S extraction and matching. NCC-S can be trivially implemented in the Similarity-MTC framework, but here we illustrate an optimized version. *Extraction*: The image (A) is first blurred with an isotropic Gaussian to remove noise. At each keypoint, pixel values are sampled on a log-polar grid (B), using a scaled image pyramid for efficiency, and stored in an $N_S \times N_R$ array (C). The array is zero padded, producing a $2N_S \times N_R$ array (D). The padded array is then mapped to Fourier space (A_F). Additionally, an array of $2N_S - 1 \times 3$ summary statistics is recorded (A_S), which will allow fast normalized cross-correlation at matching time. The final descriptor is the complex-valued Fourier array with the real-valued statistics array. *Matching*: The complex conjugate of the Fourier part of A is multiplied using the Hadamard product with the Fourier part of B, and the result is mapped to Euclidean space. This produces a $2N_S - 1 \times N_R$ array of real cross-correlation values (E). The $2N_S - 1 \times 3$ summary statistics arrays are then used to normalize the cross-correlation array on a row by row basis. The final similarity score S is the maximum value in the NCC array (F), and the offset (similarity transformation) is given by its coordinates. The final distance is defined to be $D := 1 - S$.

the matrix of ones with the same size as X and $\tilde{X}$ is X normalized to zero mean and unit Frobenius norm.

A compressed-descriptor version is also possible, where only raw pixels are stored, and the FFT and summary statistic computations are deferred to match-time.

In *optimized matching*, the method for obtaining the CC follows from the cross-correlation theorem. We then transform the CC into a NCC to account for changes in cylinder overlap and to gain illumination invariance. To do this, we express each element of the NCC in terms of an element of the CC. Suppose we observe some value u in the CC, where $u := \text{Tr}(X^\top Y)$ is the correlation (inner product) of the blocks $X, Y \in \mathbb{R}^{n_1 \times n_2}$. Let $J \in \mathbb{R}^{n_1 \times n_2}$ be the matrix of ones. Write $X = a_X \tilde{X} + b_X J$, where $b_X := \langle X_{ij} \rangle$ and $a_X := ||X - b_X J||_F$, and do the same for Y. Note

$$u = a_x a_y \text{Tr}(\tilde{X}^\top \tilde{Y}) + a_y b_x \text{Tr}(J^\top \tilde{Y}) + a_x b_y \text{Tr}(J^\top \tilde{X}) + b_x b_y \text{Tr}(J^\top J) \tag{1}$$

$$= a_x a_y \text{Tr}(\tilde{X}^\top \tilde{Y}) + a_y b_x \sum \tilde{Y}_{ij} + a_x b_y \sum \tilde{X}_{ij} + b_x b_y n_1 n_2. \tag{2}$$

Note also $\tilde{X}$ and $\tilde{Y}$ have norm one and average value zero. So, solving for the normalized correlation, we get

$$\text{Tr}(\tilde{X}^\top \tilde{Y}) = \frac{u - a_y b_x \sum \tilde{Y}_{ij} - a_x b_y \sum \tilde{X}_{ij} - b_x b_y n_1 n_2}{a_x a_y}, \tag{3}$$

thus expressing the NCC in terms of the CC and the summary statistics we collected. The
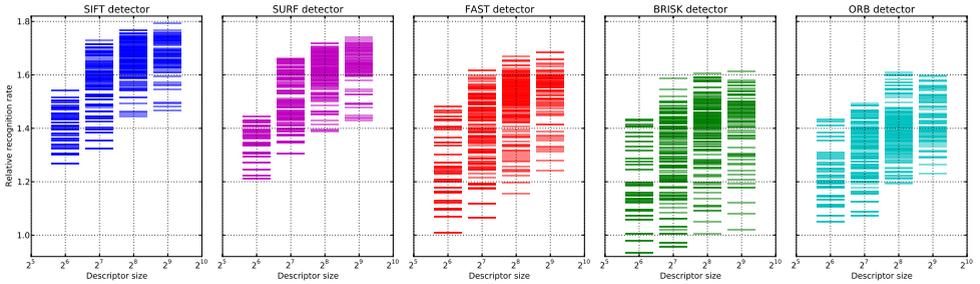
Figure 4:   A visualization of the search used to select the optimal NCC-S detector and parameters. Each horizontal line represents NCC-S matching performance for a particular choice. Performance is the relative recognition rate of NCC-S versus the average of FAST-BRIEF, SIFT, and BRISK on the `1:2`, `1:4`, and `1:6` pairs of the Oxford dataset. Typically, NCC-S accuracy increases as the number of scale levels and angle gradations increases. Here, the search identified a sweet spot, using the SIFT detector, 8 scale levels, and 16 angle gradations, corresponding to a compressed descriptor size of 128 floats. In this spot, NCC-S descriptor size is reasonable but NCC-S remains accurate.

final distance between descriptors A and B with overlapping block pairs O is

$$\Omega(A,B) := \min_{(X,Y)\in O} 1 - \text{Tr}(\tilde{X}^\top \tilde{Y}). \tag{4}$$

This distance implicitly normalizes by overlap size by normalizing each block to have mean zero and norm one.

3.2.1   Parameters

NCC-S has six extraction parameters, including the choice of keypoint detector. σ_{blur} is the standard deviation of the Gaussian kernel used to pre-blur the grayscale images. R_{min} and R_{max} are the minimum and maximum radii of the log-polar pattern. N_S and N_R are the number of scale levels and rotation gradations. It has one matching parameter: N_O, the minimum number of scale levels considered for an overlapping region in the cylinder alignment step. We had slightly better results using the l_1 distance on either rank vectors [22] or z-normed patches, but proceeded with l_2 to enable fast matching in Fourier space.

Parameter values were selected via grid search, with results visualized in Figure 4. The grid search entailed tens of thousands of experiments, run using a descriptor evaluation framework we developed using Spark [21]. The framework seamlessly scales from single-machine to cluster computations.[1] The final parameters used the SIFT detector with $\sigma_{\text{blur}} := 1.2$, $R_{\text{min}} := 4$, $R_{\text{max}} := 32$, $N_S := 8$, $N_R := 16$, and $N_O := 4$.

4   Experiments in correspondence matching

In this section, we demonstrate the efficacy of match-time covariance (MTC) for invariant descriptor construction. To do this, we test NCC-S on the Oxford [9] and Brown [19] image

---

[1]The framework is open-source and is not named here to preserve anonymity.
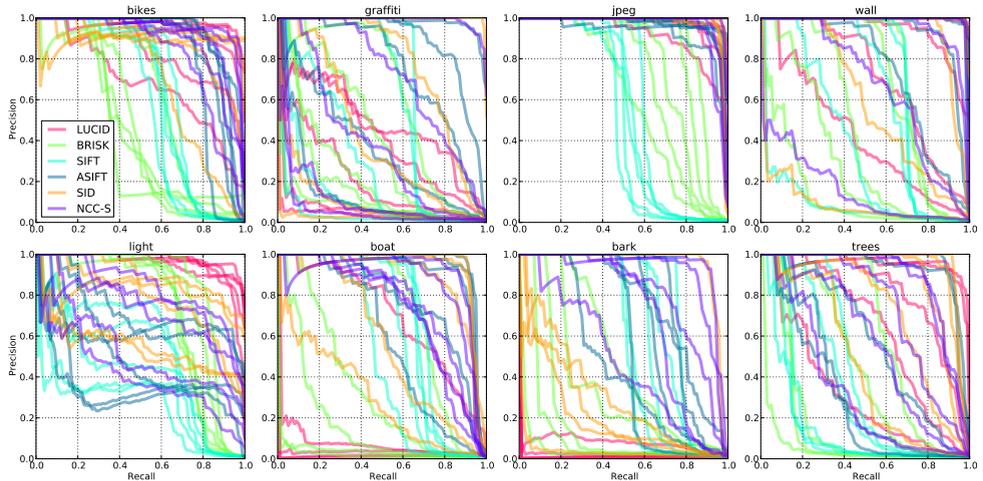
Figure 5: Precision-recall curves for various methods on the Oxford image dataset, obtained following the protocol of [9]. Each graph represents an image class. The five curves for each method represent the five image pairs `1:2` to `1:6` in each image class. Best viewed on a high-resolution color display.

datasets. We also test Similarity-MTC versions of SIFT and LUCID on the Oxford dataset. We report timings in Table 3.

4.1 NCC-S

The Oxford image dataset contains eight scenes: `boat` and `bark` have scale and rotation change. `graffiti` and `wall` have affine viewpoint change. `bikes` and `trees` have blur change. `jpeg` has change in jpeg compression. `light` has illumination change. Each scene consists of six images: a base image and 5 other images related to the base by a provided homography. The descriptor evaluation task is to match local regions in the base image to local regions in the other images [9].

We compared against five other methods. LUCID [27], a refinement of BRIEF [3], is a non-variant baseline. We implemented this simple method ourselves, using patches of size 24×24. BRISK [6] is a fast, similarity extract-time covariant (Similarity-ETC), version of BRIEF. We include it to demonstrate the gap that still exists between modern BRIEF descendants and SIFT. SIFT [8] is accurate, complicated, and uses Similarity-ETC. ASIFT [11] is slow and endows SIFT with full skew and aspect-ratio invariance through MTC, but retains SIFT's Similarity-ETC. We used OpenCV for the previous three methods. SID [5] is a rotation and scale *robust* descriptor, and is a non-MTC analog of NCC-S. We used the author's code, which produced compressed descriptors of size 1008 floats.

We report performance in Figures 5 and 6. Here, NCC-S used the parameters discussed in Section 3.2.1.

ASIFT significantly outperforms SIFT, and ASIFT is simply SIFT with skew and aspect-ratio MTC.

On the whole, NCC-S significantly outperforms all methods other than ASIFT. This includes BRISK and SIFT, most notably on the scenes with little viewpoint variation: `bikes`,
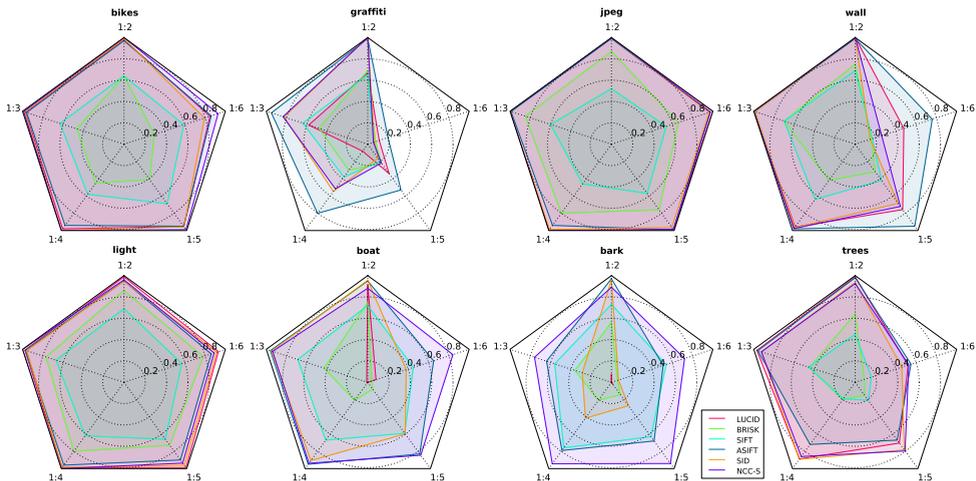
Figure 6: Recognition rates for various methods on the Oxford image dataset, obtained following the protocol of [22] and using the 100 strongest keypoints per image. Each pentagon represents an image class. Each vertex represents an image pair, ranging from `1:2` to `1:6`. Recognition rates are given by radial distance; the best methods are those which fill the pentagons. Best viewed on a high-resolution color display.

`jpeg`, `light`, and `trees`. In such cases, ETC can only add noise. Indeed, the non-variant LUCID does just as well as NCC-S. NCC-S outperforms SID for scenes with large scale change: `boat` and `bark`. This is natural, because SID lacks the true scale invariance enabled by Similarity-MTC. No winner is apparent between NCC-S and ASIFT. ASIFT's affine MTC improves its performance on scenes with pronounced affine warps: `graffiti` and `wall`. NCC-S's Similarity-MTC helps on scenes with significant rotation and scaling: `boat` and `bark`.

Table 1: Error rates for SIFT and NCC-S on the `Liberty` and `Notre Dame` categories of the Brown dataset. Error is measured at 95% recall, as in [19]. SIFT error rates are taken from [19]. NCC-S error rates are obtained using 32 scale levels and 32 rotations. These numbers show NCC-S is robust to 3D viewpoint change, despite its large support region.

The Brown dataset consists of image patches of points on 3D objects as seen from different viewpoints. This fully 3D motion model is more realistic than the homographies of the Oxford dataset, as it allows for nuisances such as occlusion. NCC-S outperforms SIFT on this dataset, as shown in Table 1.

4.2 LUCID-S and SIFT-S

Here, we test two additional descriptors: LUCID-S and SIFT-S. They are Similarity-MTC wrappings of LUCID and SIFT, respectively. In the case of SIFT-S, we disabled SIFT's scale and rotation estimation to provide pure MTC. Recognition rates are reported in Table 2.

method	SIFT	NCC-S
Liberty	0.35	0.27
NotreDame	0.26	0.22

Table 2: Recognition rates for regular and Similarity-MTC versions of LUCID and SIFT on all pairs `i:j` of the `graffiti`, `wall`, and `bark` scenes of the Oxford dataset, using the protocol used for Figure 6. They show a significant performance boost for the Similarity-MTC versions of both methods, demonstrating its general applicability. The gap between SIFT and SIFT-S also illustrates the message of Figure 2: extract-time covariance extracts a large cost.

	graffiti					wall					bark				
	1:2	1:3	1:4	1:5	1:6	1:2	1:3	1:4	1:5	1:6	1:2	1:3	1:4	1:5	1:6
LUCID	0.66	0.56	0.07	0.33	0.11	1.0	1.0	0.95	0.82	0.49	0.09	0.0	0.0	0.03	0.0
LUCID-S	0.99	0.86	0.54	0.26	0.09	1.0	1.0	0.99	0.83	0.44	0.84	0.59	0.61	0.38	0.06
SIFT	0.65	0.64	0.38	0.2	0.06	0.69	0.7	0.64	0.42	0.17	0.73	0.56	0.76	0.63	0.54
SIFT-S	1.0	0.84	0.62	0.33	0.1	0.99	0.98	0.95	0.91	0.49	0.91	0.71	0.87	0.81	0.13

Table 3: Marginal extraction and matching times for leading invariant descriptors, measured on a 16-core Xeon server running Ubuntu. NCC-S timings are for the implementation described in Figure 3. NCC-S is slower than BRISK and SIFT while offering better accuracy, and faster than ASIFT while offering comparable accuracy. Additionally, while SIFT implementations are unlikely to get faster, the simplicity and low asymptotic complexity of NCC-S suggest it should be possible to make it significantly more efficient. In particular, the NCC-S matching time is an order of magnitude longer than we expected, suggesting our code can be substantially optimized.

method	BRISK	SIFT	NCC-S	ASIFT
extraction μs	19	65	80	2700
matching ns	3	7.5	340	13000

5 Discussion

In this paper, we propose match-time covariance (MTC) as an alternative to extract-time covariance (ETC). MTC is a general strategy for descriptor design that provides *invariance* without loss of sensitivity. We also presented Similarity-MTC, a general framework for adding MTC up to similarity transformations to an existing descriptor. The framework is simple, dramatically improves accuracy vs. ETC, and can return similarity transformations. Finally, we proposed NCC-S, an accurate descriptor designed for fast execution in the Similarity-MTC framework. NCC-S is extremely simple; it is just normalized cross-correlation mapped to the Similarity-MTC framework. Its success shows it may not always be necessary to use highly complicated descriptors like SIFT or even moderately complicated descriptors like BRISK and FREAK.

MTC has several implications. *First*, it supports a notion that has gained traction in recent years, the idea that descriptors can be simple yet effective. Simplicity has a range of benefits. Practically, it means faster code with fewer bugs. Conceptually, it suggests an arrival at some atomic truth, a self-contained building block for aiding future discovery. *Second*, MTC yields rich information on the transformations between points (*offset*). Though unexplored in this paper, this information could be of use in other computer vision algorithms, e.g. it could provide additional constraints for geometric model fitting. *Third*, as with ASIFT, it is largely incompatible with lines of work built on top of local descriptors, e.g., fast nearest neighbor and feature histograms. This is because MTC is fundamentally a comparison between sets, and existing literature typically assumes descriptors are points in a vector space [4, 16].

There are concepts that lie outside linear algebra, and MTC suggests we should mind them.

Future work on NCC-S may shift it to purely integer math using an integer Fourier transform [12, 14]. This will enable fast processing on mobile devices lacking floating-point hardware. There are also many more kinds of MTC to explore, including freer motion models, blur, and illumination.

6   Acknowledgments

References

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast Retina Keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517. Ieee, June 2012. ISBN 978-1-4673-1228-8. doi: 10.1109/CVPR.2012.6247715. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6247715.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *ECCV 2006*, 2006. URL http://www.springerlink.com/index/E580H2K58434P02K.pdf.

[3] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *ECCV 2010*, 2010. URL http://www.springerlink.com/index/h8h1824827036042.pdf.

[4] M S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388. ACM, 2002. ISBN 1581134959. URL http://portal.acm.org/citation.cfm?id=509907.509965.

[5] Iasonas Kokkinos and Alan Yuille. Scale invariance without scale selection. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587798. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4587798.

[6] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. BRISK: Binary Robust invariant scalable keypoints. *2011 International Conference on Computer Vision*, pages 2548–2555, November 2011. doi: 10.1109/ICCV.2011.6126542. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6126542.

[7] JP Lewis. Fast template matching. *Vision Interface*, pages 120–123, 1995. URL http://www.idiom.com/~zilla/Work/nvisionInterface/vi95_lewis.pdf.

[8] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL http://www.springerlink.com/openurl.asp?id=doi:10.1023/B:VISI.0000029664.99615.94.

[9] Krystian Mikolajczyk and Cordelia Schmid. Performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–30, October 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.188. URL http://www.ncbi.nlm.nih.gov/pubmed/16237996.

[10] J Morel and G Yu. Is SIFT scale invariant? *Inverse Problems and Imaging*, 5 (1), 2011. URL http://dev.ipol.im/~morel/SIFT_Formalization/SIFT_Formalization_v13.pdf.

[11] Jean-Michel Morel and Guoshen Yu. ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, January 2009. ISSN 1936-4954. doi: 10.1137/080732730. URL http://epubs.siam.org/doi/abs/10.1137/080732730.

[12] T.Q. Nguyen. On the Fixed-Point Accuracy Analysis of FFT Algorithms. *IEEE Transactions on Signal Processing*, 56(10):4673–4682, October 2008. ISSN 1053-587X. doi: 10.1109/TSP.2008.924637. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4626107.

[13] Ville Ojansivu and J Heikkila. Blur invariant registration of rotated, scaled and shifted images. *The 2007 European Signal Processing Conference*, pages 1755–1759, 2007. URL http://www.ee.oulu.fi/mvg/files/pdf/EUSIPCO07.pdf.

[14] Soontorn Oraintara, Ying-Jui Chen, and Truong Q. Nguyen. Integer fast Fourier transform. *Signal Processing, IEEE Transactions on*, 50(3):607–618, 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=984749.

[15] Ethan Rublee and Vincent Rabaud. ORB: an efficient alternative to SIFT or SURF. *ICCV*, 2011. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:ORB+:+an+efficient+alternative+to+SIFT+or+SURF#0http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6126544.

[16] AWM Smeulders and Marcel Worring. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349–1380, 2000. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=895972.

[17] Tinne Tuytelaars and Krystian Mikolajczyk. Local Invariant Feature Detectors: A Survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2007. ISSN 1572-2740. doi: 10.1561/0600000017. URL http://www.nowpublishers.com/product.aspx?product=CGV&doi=0600000017.

[18] A. Vedaldi and S. Soatto. Features for recognition: Viewpoint invariance for non-planar scenes. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 1474–1481 Vol. 2, 2005. doi: 10.1109/ICCV.2005.99. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1544892.

[19] S. Winder and M. Brown. Picking the best DAISY. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 178–185, June 2009. doi: 10.1109/CVPR.2009.5206839. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206839.

[20] George Wolberg and S Zokai. Robust image registration using log-polar transform. *Image Processing, 2000. Proceedings. 2000 International Conference on*, 2000. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=901003.

[21] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, and Ankur Dave. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, 2012. URL https://www.usenix.org/system/files/conference/nsdi12/nsdi12-final138.pdf.

[22] Andrew Ziegler, Eric Christiansen, David Kriegman, and Serge Belongie. Locally Uniform Comparison Image Descriptor. *Advances in Neural Information Processing Systems 25*, pages 1–9, 2012. URL http://books.nips.cc/papers/files/nips25/NIPS2012_0012.pdf.