Unified Real-Time Tracking and Recognition with Rotation-Invariant Fast Features

Gabriel Takacs[†] Vijay Chandrasekhar[†] Sam Tsai[†]
David Chen[†] Radek Grzeszczuk[‡] Bernd Girod[†]

[†] Information Systems Laboratory
Stanford University
{gtakacs,vijayc,sstsai,dmchen,bgirod}@stanford.edu

[‡] Nokia Research Center
Palo Alto, CA
radek.grzeszczuk@nokia.com

Abstract

We present a method that unifies tracking and video content recognition with applications to Mobile Augmented Reality (MAR). We introduce the Radial Gradient Transform (RGT) and an approximate RGT, yielding the Rotation-Invariant, Fast Feature (RIFF) descriptor. We demonstrate that RIFF is fast enough for real-time tracking, while robust enough for large scale retrieval tasks. At 26× the speed, our tracking-scheme obtains a more accurate global affine motion-model than the Kanade Lucas Tomasi (KLT) tracker. The same descriptors can achieve 94% retrieval accuracy from a database of 10^4 *images.*

1. Introduction

Mobile Augmented Reality (MAR) systems rely on two key technologies, visual tracking and recognition. These two components are often used as complementary pairs, where periodic recognition results are bridged by tracking the recognized content. Some MAR systems, such as Taylor *et al.* [1], have eliminated tracking by performing recognition at sufficiently high frame rates. However, such approaches inherently suffer from temporal coherency challenges, and large memory requirements. If neighboring frames do not return consistent results then jitter may occur. Additionally, at video frame rates, the redundancy between successive frames is extremely large. Consequently, it is inefficient to perform a full recognition on every frame.

In this work we aim to exploit the redundancy between frames for unifying tracking and recognition. To do so we must extract information at video frame rates that is useful for both recognition and tracking. Achieving such frame rates can be quite challenging on handheld devices used in many MAR systems. We solve this challenge by designing a local feature descriptor that is both robust and extremely fast to compute. Very little effort is then needed to achieve both tracking and recognition.

1.1. Prior Work

With the introduction of highly capable smartphones, there has been much recent interest in combining tracking and content recognition for MAR [2, 3, 4, 5, 6, 1]. Systems have been proposed to track video content using feature descriptors, including motion vector tracking [7], SURFTrac [2], and patch tracking [5]. There have been numerous systems for image and video content recognition using natural feature descriptors. Many of these systems rely on a Histogram of Gradients (HoG) descriptor similar to SIFT [8], SURF [9], DAISY [10], or CHoG [11].

For this work, we are interested in systems which perform simultaneous tracking and recognition. Wagner *et al.* [4, 5, 6] have made significant progress towards such systems on mobile phones. However, their systems use different methods for tracking (PatchTracker) and for recognition (PhonySIFT or PhonyFERNS). Klein and Murray [12] have implemented parallel tracking and mapping on a mobile phone, which allows them to track and localize points in space. However, their system does not recognize content. Ideally, the same data could be used for tracking and recognition, however, most prior recognition systems are prohibitively slow to also use for tracking.

The work by Taylor [13, 1] has largely eliminated tracking by applying a high-speed recognition system at video frame rates. However, their system requires a large amount of memory and database redundancy, and is not yet optimized for real-time operation on hand-

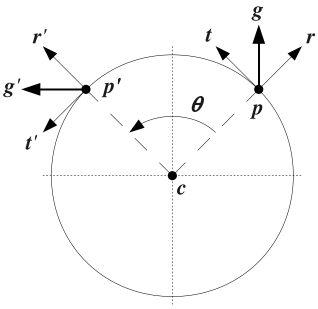


Figure 1. Illustration of radial gradients. The gradient, g, is projected onto a local, radial coordinate system (r, t). When the interest point, c, is rotated by θ the radial gradients are the same.

held devices. More importantly, it is wasteful to perform full content recognition on every frame of a video stream because adjacent frames are highly correlated.

For unification, we need a descriptor that can be computed at video frame rates on a handheld device. Therefore, we wish to have a very simple descriptor, while maintaining state-of-the-art robustness. Others have proposed extremely fast descriptors [14, 1], however, the robustness of these descriptors is sacrificed. As a starting point, we use our prior work on CHoG [11], which has been shown to perform well at very low bitrates. For speed, we remove the orientation assignment phase of keypoint detection, and instead propose a Rotation-Invariant, Fast Feature (RIFF).

There are two prominent techniques for rotation invariance in the current literature. The first is using steerable filters, often with descriptor permutation [10, 15]. This method suffers from high computational overhead of computing many filter orientations. The second technique is to treat rotation as a circular shift and use the magnitude of the Fourier transform [16, 17]. This method is often not sufficiently robust to viewpoint variation. An additional technique has been propsed by Brasnett and Bober [18], and included in the MPEG-7 Image Signatures standard. Their method computes scalar statistics over circular regions. Though these signatures are rotation invariant, they are not robust to viewpoint variation.

1.2. Contributions

In this work we present the RIFF descriptor, which provides state-of-the-art robustness and speed. We do so by presenting a gradient transform that is extremely simple to compute, and leverages on the proven methods of SIFT and CHoG. We then use RIFF for tracking in real-time on a handheld device. Our tracker provides more accurate results than the KLT tracker [19], at 26× the speed. We show that RIFF is also capable of large-scale retrieval tasks with up to 10^6 images.

In the rest of the paper, we first introduce and evaluate the RIFF descriptor in Section 2. We then show how to use RIFF for real-time tracking in Section 3, and evaluate the tracker's performance in Section 4. Finally, in Section 5, we demonstrate how RIFF tracking can be used with content recognition systems.

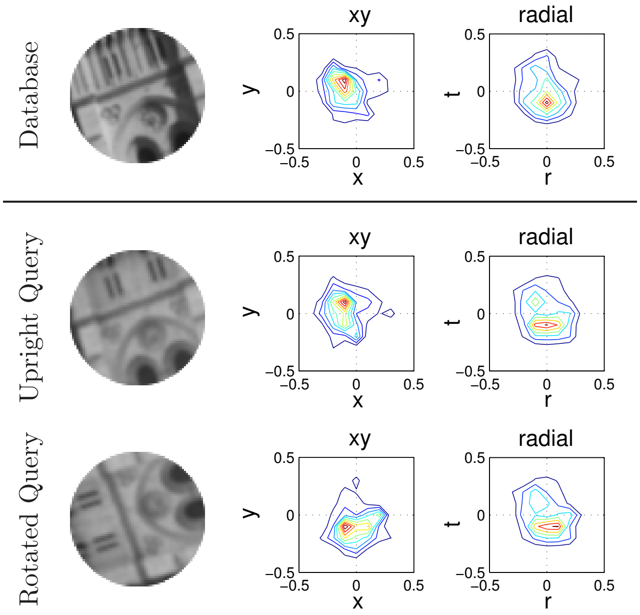


Figure 2. Illustration of rotation invariance with the RGT. A database patch (*top left*) produces a gradient histogram in both the xy (*top middle*) and radial (*top right*) domains. Similar histograms are extracted from an upright query patch (*center left*) and its rotated version (*bottom left*). We note that the xy-gradient histograms (*center column*) rotate with the patch, while the radial-gradient histograms (*right column*) maintain the same shape across all rows.

2. Rotation-Invariant Descriptors

Any image recognition algorithm for handheld devices must be rotationally invariant. Typical feature descriptor based systems, such as SIFT [8] and SURF [9], assign an orientation to interest points before extracting descriptors. Some systems, such as [15], use reorientable descriptors, and others, such as MPEG-7 [18], use rotation invariant transforms.

To design a feature descriptor that is fast enough to use at video frame rates on handheld devices we must simplify the computation as much as possible. To achieve 30 frames per second (fps) with 640×480 frames on a 600 MHz processor, we can only expend 65 cycles per pixel. In practice, there are even fewer cycles available to the MAR application. Because of this constraint, we choose to use orientation invariant descriptors which eliminate the computation of finding an orientation and interpolating the relevant pixels.

2.1. Gradient Binning

To design RIFF, we start with a HoG type descriptor and identify the two parts of the pipeline that are not rotation-invariant: spatial binning and gradient binning. To make gradient binning invariant we apply an invertible, spatially-varying transform. By rotating the gradients to the proper angle, we achieve rotation invariance with no loss of information, yielding the Radial Gradient Transform (RGT).

As shown in Figure 1, we choose two orthogonal basis vectors to provide a local, polar reference-frame for describing the gradient. These basis vectors, r and t, are the radial and tangential directions at a point p, relative to the center of the patch, c. We define R_θ as the rotation matrix for angle θ, yielding

$$r = \frac{p - c}{\|p - c\|}, \qquad t = R_{\frac{\pi}{2}} r \tag{1}$$

By projecting onto r and t, we can decompose g into its local coordinate system as $(g^\mathrm{T} r)\, r + (g^\mathrm{T} t)\, t$, such that the gradient can be represented in the local radial coordinate system as the vector $(g^\mathrm{T} r, g^\mathrm{T} t)$. Now assume that the patch has been rotated about its center by some angle, θ. This yields a new local coordinate system and gradient

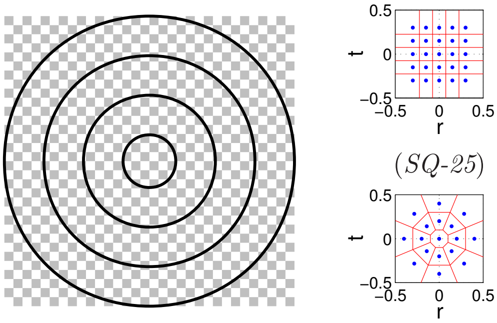
$$R_\theta p = p', \quad R_\theta r = r', \quad R_\theta t = t', \quad R_\theta g = g'.$$

The coordinates of the gradient in the local frame are invariant to rotation, which is easily verified by

$$\begin{aligned} \left({g'}^\mathrm{T} r', {g'}^\mathrm{T} t'\right) &= \left((R_\theta g)^\mathrm{T} R_\theta r, (R_\theta g)^\mathrm{T} R_\theta t\right) \\ &= \left(g^\mathrm{T} R_\theta^\mathrm{T} R_\theta r, g^\mathrm{T} R_\theta^\mathrm{T} R_\theta t\right) \\ &= \left(g^\mathrm{T} r, g^\mathrm{T} t\right) \end{aligned} \tag{2}$$

All gradients are rotated by the same angle and R_θ is a one-to-one mapping. Thus, the set of gradients on any given circle centered around the patch is invariant to rotation, as shown in Figure 2.

Given these rotation-invariant gradients, we apply a binning technique, as in CHoG [11], to produce histograms of gradients which comprise the descriptor. We bin the gradient histogram to reduce its dimensionality, while maintaining robustness. Figure 3 shows the gradient binning centers and Voronoi cells which are used for RIFF. We show a vector quantizer, *VQ-17*, and a scalar quantizer, *SQ-25*. Vector quantization is flexible and can be matched to the gradient statistics, while scalar quantization is extremely fast.



(*Annuli*) (*SQ-25*) (*VQ-17*)

Figure 3. With our spatial binning configuration, *Annuli*, we can gain a 2× speedup by only sampling the gray pixels. Also shown are the bin centers (*blue*) and Voronoi cells (*red*) for histogram quantization. The vector quantizer (*VQ-17*) is more flexible, while the scalar quantizer (*SQ-25*) is faster.

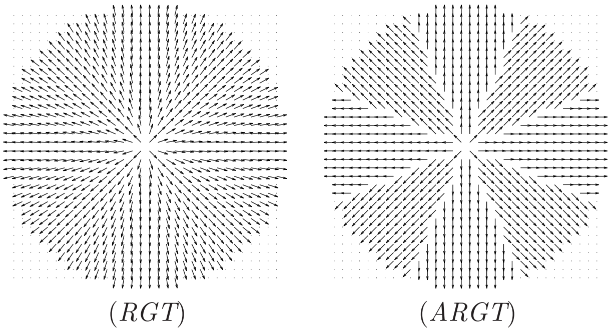
2.2. Spatial Binning

Now that we have discussed rotation-invariant gradient binning, we consider how to perform spatial binning. A single gradient histogram from a circular patch would be rotation-invariant, but would not be sufficiently distinct. To improve distinctiveness while maintaining rotation invariance, we subdivide the patch into annular spatial bins. An example of this *Annuli* spatial binning configuration is shown in Figure 3. Note that there is a trade-off in performance between the number of annuli and their width. More spatial bins increases distinctiveness, but narrower annuli decreases robustness. We typically use four annuli with a total diameter of 40 pixels.

2.3. Speed Enhancements

We now discuss how to enhance the speed of our proposed descriptors to make them amenable to real-time tracking on a handheld device. Since we have removed the orientation assignment stage, we can simply extract the descriptor from an upright patch around the interest point. This eliminates the costly step of interpolating pixels.

Approximate RGT. A naïve implementation of the RGT would require a large number of costly floating-point matrix multiplications. Even using fixed-point arithmetic to speed up the computation would add significant complexity to the descriptor. However, the exact r and t basis vectors can be approximated by a simpler pair, $\hat{r}$ and $\hat{t}$. As seen in Figure 4, these approximate basis vectors are quantized to 45°, which allows us to directly compute the gradient along that direction with no additional cost. This gives us an Approximate



(RGT) $(ARGT)$

Figure 4. Illustration of the RGT basis vectors (*left*), and the ARGT basis vectors (*right*). Gradients along the ARGT bases can be efficiently computed directly from pixel data.

Radial Gradient Transform (ARGT) that is computed by finding the differences between neighboring pixels with the appropriate normalization.

Histogram Estimation. Local HoG descriptors capture the statistics of image content around an interest point. Assuming the image content of two interest points is the same, then the distribution of gradients should be similar. This underlying distribution is estimated by a histogram of samples, with more samples leading to a better estimate. However, each sample requires computing and quantizing the gradient. Hence, there is a trade-off between the speed of computation and the quality of the estimate, via the number of samples. We can improve the speed, with minor degradation to the estimate, by subsampling pixels around the interest point. Such a scheme is shown in Figure 3, where the gray pixels are used to estimate the HoG.

Scalar Quantization. Once we have computed the ARGT, we quantize the gradient histograms to construct our descriptor. We do so in the same way as CHoG [11], however, general vector quantization can be too slow for real-time performance on a handheld device. This is because gradient binning is performed an the inner most loop of the RIFF algorithm. To meet our tight speed constraints, we use a 5×5 scalar quantizer in place of a vector quantizer. This yields a 100-dimensional RIFF descriptor.

2.4. Descriptor Evaluation

The best way to compare feature descriptors without the rest of the image matching apparatus is to use the ROC curve. We generate ROC curves using the methods of Winder *et al.* [15], using their *Liberty* dataset. The resulting ROC curves are shown in Figure 5. The RIFF descriptor suffers some performance loss relative to SIFT due to fewer spatial bins. Since the SIFT descriptor is not orientation invariant, we also compare against MPEG-7 Image Signatures, which perform significantly worse than RIFF. Note that there is no performance loss by using the ARGT instead of the RGT.

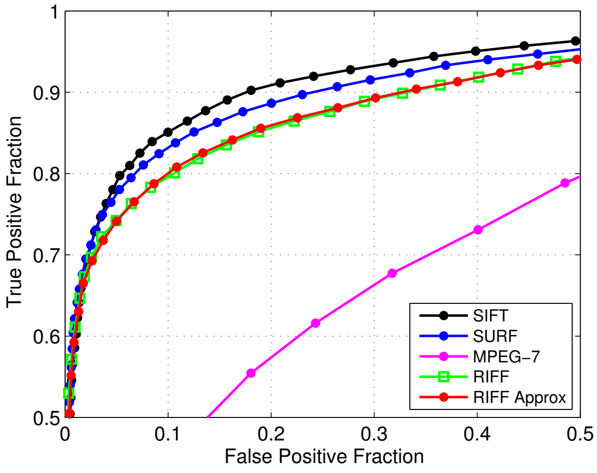


Figure 5. Receiver operating characteristics for the *Liberty* dataset. RIFF underperforms SIFT, but performs significantly better than MPEG-7 Image Signatures. Approximating the radial gradients does not effect performance.

3. Tracking

Given an extremely fast, high-quality feature descriptor, we can combine tracking and recognition by using the same descriptors for both tasks. This section discusses how we use these descriptors for tracking.

First, we apply the FAST [20] interest point detector on each level of an image pyramid. For speed we omit non-integer levels of the pyramid, as in [13]. This provides good coverage in scale space, while not requiring any pixel interpolation beyond 2× downsampling. When used for recognition, any lack of coverage in scale space can be overcome by redundancy in the database.

These descriptors are then matched to spatially neighboring descriptors in the previous frame. If two descriptors fall within a fixed radius (8 pixels) then they are considered candidates for a match. The best candidate is then chosen as that with the minimum distance in the descriptor domain, subject to a distance threshold. We use KL-divergence, which provides the best matching performance, as shown in [11].

To enable descriptor matching at very high frame rates, we use a fast hashing and spatial binning of matching candidates according to their location in the frame. We divide the frame into a spatial grid and place the current frame's descriptors into the bin from which

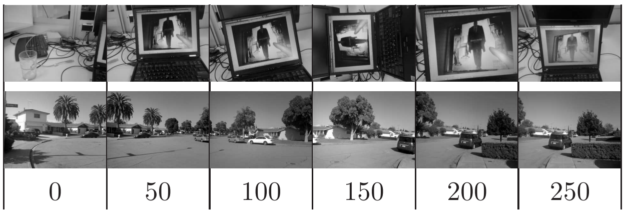


Figure 6. Sample frames from the *Laptop* (*top*) and *Street* (*bottom*) videos. *Laptop* contains pan, zoom, and rotation, while *Street* only contains panning. The videos are 320×240 at 15 fps. The frame number is listed below the frame.

it was detected, as well as into the eight neighboring bins. This binning allows for fast lookup of spatial neighbors between frames. To determine matching candidates, we simply find the bin into which the current frame's descriptor falls. This bin contains a short list of all neighboring descriptors from the previous frame.

This matching technique provides feature matches that are sufficiently robust and free of outliers such that no outlier removal is required. To track the global movement of the frame, we compute a least-squares solution to an affine model between the current and previous frames. The accuracy of this affine model is discussed in Section 4.

At no additional computational cost, we compile a backlog of previous frames, their matches, and affine models. Having this temporally dense information allows us to achieve quality recognition with a very modest number of descriptors per frame.

4. Tracking Evaluation

To evaluate the performance of our tracking scheme, we compare it with a standard implementation of the KLT tracker [19], which detects interest points and tracks them using image intensity. We compare the tracker's speed and its resulting global affine models. For both RIFF and KLT, we track 100 features through two 15 fps QVGA videos of a laptop and a street. These videos are illustrated in Figure 6. To compare the speed, we ran both trackers on a handheld and laptop computer. The handheld is a Nokia N900 with a 600 MHz ARM processor, and the laptop is an IBM T43 with a 1.8 GHz Pentium M processor.

Figure 7 plots the global affine model resulting from tracking the *Laptop* dataset. The model is relative to the first frame. The top plot shows the offset parameters of the affine model. The bottom plot shows the linear portions of the affine model, where A_{11} and A_{22} are the diagonal elements. We see that the model starts off as the identity matrix, then pans right and down before zooming in while rotating. Then, at frame 150, the off-diagonals approach ±1 while the diagonals approach 0, indicating a 90° rotation. The camera motion then essentially reverses itself.

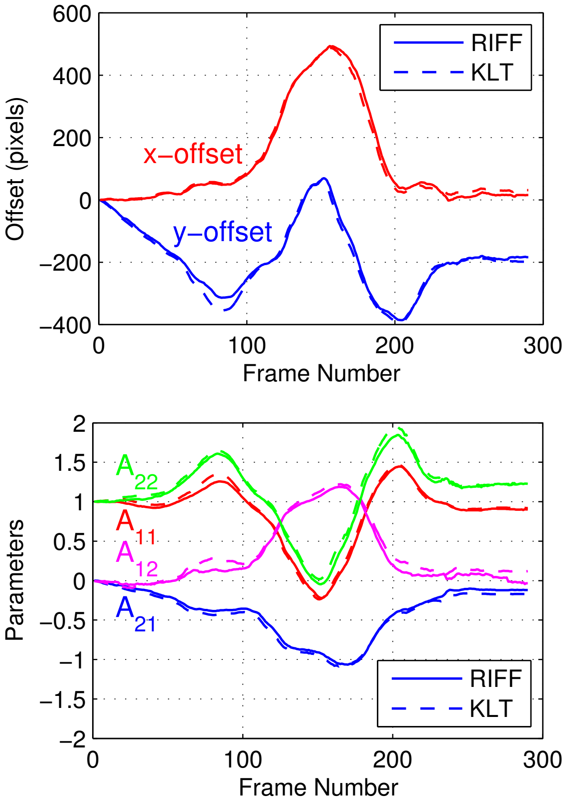


Figure 7. Affine model, relative to the first frame, for both KLT (*dashed*) and RIFF (*solid*) tracking with the *Laptop* sequence. The top plot shows the offset parameters of the model, and the bottom plot shows the linear portion.

We note that the difference between the models obtained by the KLT and RIFF trackers is very small. This shows that RIFF tracking provides global affine models that can be used in place of those provided by the KLT tracker. Additionally, we would not expect the models to be identical, as the two trackers use different interest point detectors, and thus track different points. Since the scene content is 3D, tracking different points will lead to different affine models.

To determine the accuracy of each tracker, we created palindromic versions of *Laptop* and *Street*, where the video is played forward and then backward. Any point tracked through such a video should return to its starting position. Figure 8 shows the start and end positions of a rectangle tracked with RIFF and KLT. In both sequences, RIFF is more accurate than KLT.

Figure 9 shows the speed in fps for both tracking schemes on a laptop and handheld computer. For these

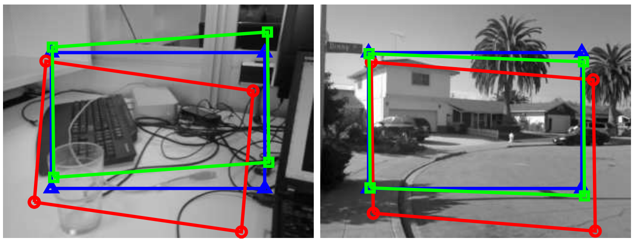


Figure 8. Tracking results from palindromic versions of *Laptop* and *Street*. A rectangle from the first frame (*blue*) is tracked forward and backward through the sequence, and should end up in the same place in the final frame. The final RIFF rectangle (*green*) is closer to the original than the KLT rectangle (*red*).

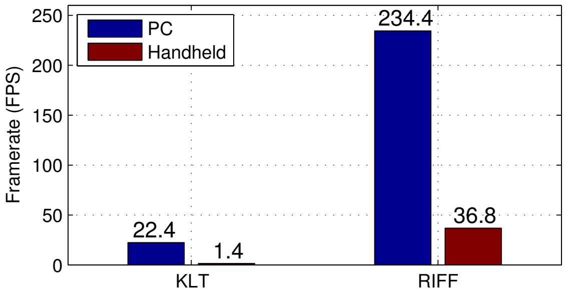


Figure 9. Frame rate for KLT and RIFF on a PC and Handheld computer. RIFF is faster than KLT by 10× on PC, and 26× on Handheld.

timings, the frames were preloaded into memory, and thus no disk or camera operations are included. We see that on the PC, RIFF tracking is 10× faster than KLT tracking, and this speedup is amplified to 26× on the handheld. The slow performance of KLT on the handheld is likely due to floating point arithmetic. Note that RIFF tracking achieves 36 fps on the handheld, which leaves enough time for camera operations and recognition while maintaining real-time performance.

5. Recognition

Using RIFF tracking, we can perform real-time feature descriptor extraction and tracking on a handheld device. Additionally, we have a buffer of past tracked features and global affine models. This means that, even though we only extract 100 features per frame, over the course of one second at 15 fps we have extracted and tracked 1500 features. This provides sufficient information for video content recognition.

The unification of tracking and recognition has the additional benefit of providing temporal coherence to the recognition data. We can infer the robustness of feature descriptors by examining their path through the video stream. This information can be used for pruning irrelevant data from the query features. In addition to pruning spurious interest points, we can improve the description of a robust interest point by obtaining more samples from neighboring frames.

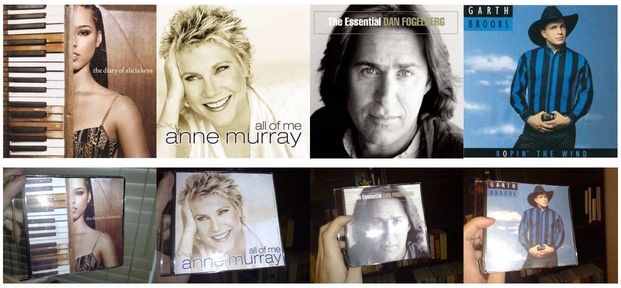


Figure 10. Example image pairs from the *CD* dataset. A clean database picture (*top*) is matched against a real-world picture (*bottom*) at various orientations.

For many MAR applications we would like to periodically query the video stream against a local or remote database. The querying may be done at a regular interval (say 1 Hz) or only when significant new content is present, which can be readily inferred from the tracking data. For outdoor MAR applications, such as [3], we may use GPS information to prefetch an appropriate local database, thus limiting the size of the database to a few thousand. For MAR applications with larger databases, such as CD recognition [21], the tracking data can be compressed and queried to a server.

In this section, we show how ROC performance translates into pairwise image matching and retrieval performance. Both experiments use images from a database of 10^6 CD/DVD/book cover images [22]. Sample images from this *CD* dataset are shown in Figure 10. Note that the 500×500-pixel database images are clean, while the 640×480-pixel queries contain perspective distortion, background clutter, and glare.

5.1. Pairwise Matching

We wish to test not only the matching performance, but also the rotational invariance of our descriptor. To do so, the query images are rotated in 5° increments, and matched against the corresponding upright database images. We use a ratio-test followed by RANSAC to ensure that the resulting feature matches are correct. RIFF and SIFT use the same Difference of Gaussian (DoG) interest points, while SURF uses fast Hessian points. The matching results are shown in Figure 11, where we plot the average number of feature matches versus the amount of rotation, averaged over 10 image pairs.

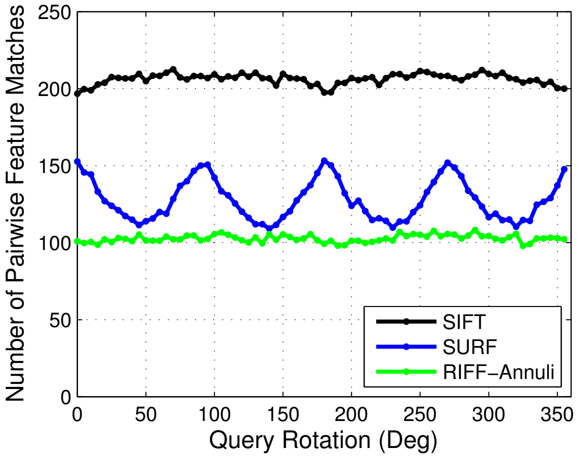


Figure 11. Pairwise matching at various image rotations for the *CD* dataset. RIFF performs comparably to SURF, and is completely rotation invariant. The SURF interest point detector causes significant variance with orientation.

From these results, we see that the DoG interest point detector is nearly isotropic, leading to a flat response for all of the schemes using it. For reference, we have included the SURF pipeline which suffers from significant anisotropy. This is due to the box filtering used in the interest point detector. As shown by the ROC performance, RIFF performs comparably to SURF. Both proposed descriptors are orientation invariant, as seen by the flat response.

5.2. Database Retrieval

For retrieval evaluation, we vary the database size from 2000 to 10^6 images, and use 1000 query images. We measure the retrieval error rate as the percentage of query images not correctly retrieved and verified with our pipeline. We briefly describe our retrieval pipeline, which is similar to other the state-of-the-art systems, such as [23, 24, 25, 26].

We first extract about 600 descriptors from DoG interest points in each image. Using these descriptors, we train a 10^6 leaf, 6 level, vocabulary tree [23]. We use symmetric KL-divergence as the distance measure for both training and querying, since it performs better than L_2-norm for HoG descriptors [27]. KL-divergence can be incorporated into the k-means clustering framework because it is a Bregman divergence [28]. For more robustness, we use soft-assignment of descriptors to the 3 nearest centroids, as in [25].

We compute a similarity between the query and database features using the standard Term Frequency-Inverse Document Frequency (TF-IDF) scheme that represents query and database images as sparse vectors of visual word occurences. We also use the weighting scheme proposed by Nistér [23], which reduces the contribution of less descriminative descriptors. Once the best 50 images are selected from the TF-IDF voting we perform pairwise matching with a ratio-test and a geometric consistency check.

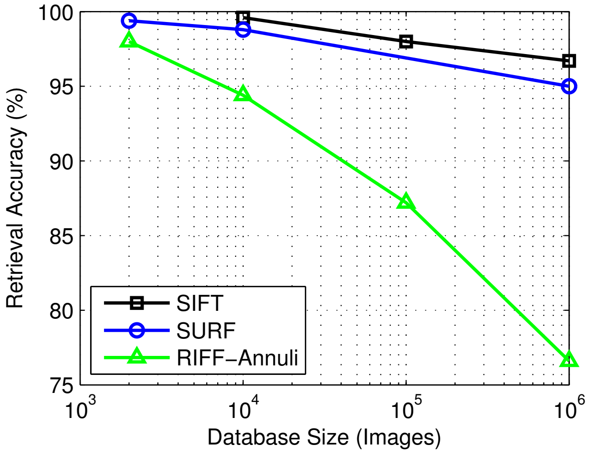


Figure 12. Accuracy of image retrieval from a database of CD, DVD, and book covers. The RIFF descriptor performs well at the retrieval task for all size databases, though accuracy declines as the database size gets very large.

Figure 12 shows the results of database retrieval using RIFF-*Annuli*, SURF, and SIFT. SIFT outperforms SURF and RIFF at all database sizes, maintaining 96% accuracy up to 10^6 images. SURF performs slightly worse, dropping to 95% at 10^6 images, while RIFF-*Annuli* achieves 77% accuracy at 10^6 images. Note that the 94% accuracy of RIFF at 10^4 images is sufficient for many recognition tasks.

6. Conclusions

We have presented a method that unifies tracking and video content recognition for MAR applications. We have introduced the RGT and its approximation, which yielded the RIFF descriptor. We have demonstrated that RIFF is fast enough for real-time tracking, and robust enough for large scale retrieval tasks. We showed that RIFF achieves 94% retrieval accuracy from a database of 10^4 images. Our tracking scheme obtains a more accurate global motion-model than the KLT, at 26× the speed. These models and feature matches can increase the robustness of video content recognition. By coupling tracking and recognition, each part of the system can take mutual advantage of each other in a natural way. We expect future MAR applications to benefit from such a unification.

References

[1] Simon Taylor, Edward Rosten, Tom Drummond, "Robust Feature Matching in 2.3us," in *Conference on Computer Vision and Pattern Recognition*, June 2009.

[2] Ta, D.-N., Chen, W.-C., Gelfand, N., Pulli, K., "SURFTrac: Efficient Tracking and Continuous Object Recognition using Local Feature Descriptors," in *CVPR*, 2009.

[3] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *ACM MIR*, Vancouver, Canada, Oct 2008.

[4] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, Dieter Schmalstieg, "Real Time Detection and Tracking for Augmented Reality on Mobile Phones," *Visualization and Computer Graphics*, Aug 2009.

[5] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose Tracking from Natural Features on Mobile Phones," in *ISMAR*, Cambridge, UK, Sept 2008.

[6] D. Wagner, D. Schmalstieg, H. Bischof, "Multiple target detection and tracking with guaranteed framerates on mobile phones," in *ISMAR*, Orlando, FL, USA, 2009.

[7] G. Takacs, V. Chandrasekhar, B. Girod, and R. Grzeszczuk, "Feature Tracking for Mobile Augmented Reality Using Video Coder Motion Vectors," in *ISMAR*, 2007.

[8] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[9] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," in *Proc. of European Conference on Computer Vision (ECCV)*, Graz, Austria, May 2006.

[10] Engin Tola, Vincent Lepetit, Pascal Fua, "A Fast Local Descriptor for Dense Matching," in *CVPR*, 2008.

[11] Vijay Chandrasekhar, Gabriel Takacs, and David Chen, Sam Tsai, Radek Grzeszczuk, Bernd Girod, "CHoG: Compressed Histogram of Gradients, a Low Bitrate Descriptor," in *CVPR*, 2009.

[12] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *ISMAR*, Orlando, October 2009.

[13] Simon Taylor, Tom Drummond, "Multiple target localisation at over 100 FPS," in *British Machine Vision Conference*, London, UK, Sept 2009.

[14] Michael Calonder, Vicent Lepetit, Pascal Fua, Kurt Konolige, James Bowman, Patrick Mihelich, "Compact Signatures for High-speed Interest Point Description and Matching," in *Int. Conf. on Computer Vision (ICCV)*, 2009.

[15] Simon Winder, Gang Hua, Matthew Brown, "Picking the best DAISY," in *Conference on Computer Vision and Pattern Recognition*, 2009.

[16] Ahonen T, Matas J, He C, Pietikinen M, "Rotation invariant image description with local binary pattern histogram fourier features," in *Image Analysis, SCIA 2009 Proceedings, Lecture Notes in Computer Science 5575*, 2009.

[17] Nick Kingsbury, "Rotation-Invariant Local Feature Matching with Complex Wavelets," in *Proc. European Conf. Signal Processing (EUSIPCO)*, 2006.

[18] P. Brasnett and M. Bober, "A Robust Visual Identifier Using the Trace Transform," in *VIE*, Jul 2007.

[19] Stan Birchfield, *KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker*, 2007. [Online]. Available: http://www.ces.clemson.edu/ stb/klt

[20] E. Rosten and T. Drummond, "Machine Learning for High Speed Corner Detection," in *9th Euproean Conference on Computer Vision*, vol. 1, Apr 2006, p. 430443.

[21] S.S.Tsai, D. Chen, J. Singh, and B. Girod, "Rate Efficient Real Time CD Cover Recognition on a Camera Phone," in *ACM Multimedia*, Vancouver, Canada, October 2008.

[22] D. M. Chen, S. S. Tsai, R. Vedantham, R. Grzeszczuk, and B. Girod, *CD Cover Database: Query Images*, April 2008. [Online]. Available: http://vcui2.nokiapaloalto.com/ dchen/cibr/testimages

[23] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *CVPR*, New York, USA, June 2006.

[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," in *CVPR*, Minneapolis, Minnesota, 2007.

[25] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, "Lost in quantization - Improving particular object retrieval in large scale image databases," in *CVPR*, Anchorage, Alaska, June 2008.

[26] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, Berlin, Heidelberg, 2008, pp. 304–317.

[27] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, R. Grzeszczuk, and B. Girod, "CHoG: Compressed Histogram of Gradients - A low bit rate feature descriptor," in *CVPR*, Miami, Florida, June 2009.

[28] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with bregman divergences," in *Journal of Machine Learning Research*, 2004, pp. 234–245.