

Fast Computation of Rotation-Invariant Image Features by an Approximate Radial Gradient Transform

Gabriel Takacs, Vijay Chandrasekhar, Sam S. Tsai, David Chen, Radek Grzeszczuk, and Bernd Girod

Abstract—We present the radial gradient transform (RGT) and a fast approximation, the approximate RGT (ARGT). We analyze the effects of the approximation on gradient quantization and histogramming. The ARGT is incorporated into the rotation-invariant fast feature (RIFF) algorithm. We demonstrate that, using the ARGT, RIFF extracts features 16x faster than SURF while achieving a similar performance for image matching and retrieval.

Index Terms—Computer vision, signal processing, real-time systems, feature representation, invariants.

I. INTRODUCTION

IMAGE matching and image-based retrieval have become central to many applications, including Augmented Reality (AR) [1], [2], set-top-box video processing [3]–[6], and web-scale image-based retrieval [7]–[9]. Many such tasks require invariance to image rotation. For example, the orientation of the camera relative to the scene may be unknown in AR applications, or, for set-top-box processing, we may wish to detect objects with unknown poses in the video.

Currently, the dominant technique for rotation-invariant image matching is to find corresponding local image features, each of which is rotation invariant. Many algorithms have been proposed for such local features, including grayscale invariants [10], SIFT [11], SURF [12], GLOH [13], CHoG [14], CARD [15], ORB [16], BRISK [17], and complex dual-tree wavelets [18]. The most accurate and robust of these techniques are based on Histograms of Oriented Gradients (HOG) [19]. As shown by Chandrasekhar *et al.* [20], SIFT and SURF descriptors are statistics from the HOG, while CHoG directly captures a coarsely quantized HOG. However, in each of these cases, the gradients must be oriented for the descriptor to be rotation-invariant. We simplify this approach by computing gradients such that they are intrinsically rotation-invariant.

In addition to local features, others have proposed rotation-invariant, global image descriptors. For example, Brasnett and

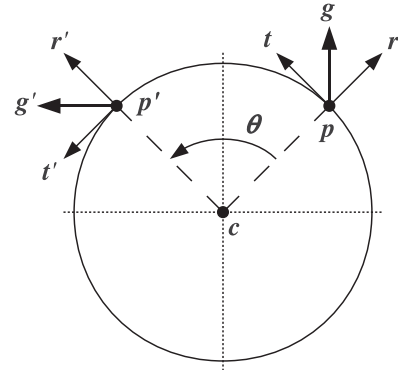


Fig. 1. Illustration of radial gradients. The gradient, g , is presented in a local, radial coordinate system (r, t) . When the local patch is rotated by θ about its center, c , the radial gradients remain unchanged.

Bober [21] use the trace transform to get a global image descriptor that is compact, rotation-invariant, and robust to small image distortions. Their method is similar to other transform methods, such as those of Roover *et al.* [22], and Seo *et al.* [23]. In these methods, a Radon (or similar) transform converts image rotation into a shift in the transform domain. The algorithm then achieves invariance to this shift using the Fourier transform. Similarly, the complex dual-tree wavelet approach proposed by Kingsbury *et al.* [18] converts rotation into a circular shift and matches features in the Fourier domain.

In this work, we first introduce the Radial Gradient Transform (RGT) and a fast Approximate RGT (ARGT) in Section II. Then, in Section III, we present a detailed analysis of the effects of the approximation on gradient quantization and histogramming. Following this analysis, Section IV describes how to utilize the ARGT in the Rotation-Invariant Fast Feature (RIFF) algorithm. We then optimize the gradient quantization parameters in Section V. Finally, in Section VI, we demonstrate the performance of RIFF for large-scale image matching and image-based retrieval.

II. RADIAL GRADIENT TRANSFORM

A. Exact Radial Gradient Transform

The RGT is an invertible, spatially-varying transform, which by rotating the gradients to the proper angle, achieves rotation invariance with no loss of information. As shown in Fig. 1, the RGT uses two orthogonal basis vectors to provide a local,

Manuscript received June 29, 2012; revised October 16, 2012; accepted November 1, 2012. Date of publication November 29, 2012; date of current version May 24, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Chun-Shien Lu.

G. Takacs and R. Grzeszczuk are with Microsoft Corporation, Sunnyvale, CA 94089 USA (e-mail: gatakacs@microsoft.com; radekgrz@microsoft.com).

V. Chandrasekhar, S. S. Tsai, D. Chen, and B. Girod are with Stanford University, Stanford, CA 94305 USA (e-mail: vijayc@stanford.edu; sstai@stanford.edu; dmchen@stanford.edu; bgirod@stanford.edu).

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.2012.2230011

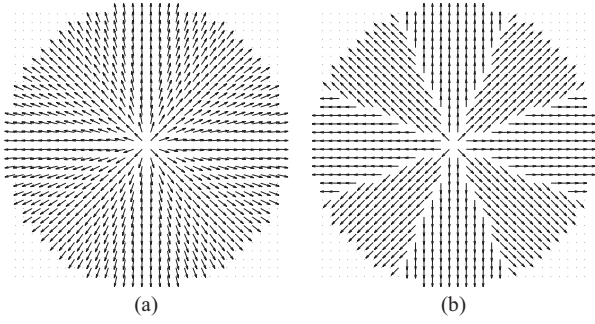


Fig. 2. (a) Illustration of the RGT basis vectors from 2. (b) ARGT basis vectors from 7. Gradients along the ARGT bases can be efficiently computed directly from pixel data.

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_α as the rotation matrix for angle α , yielding

$$r = \frac{p - c}{\|p - c\|}, \quad t = R_{\frac{\pi}{2}} r. \quad (1)$$

Alternatively, we can describe these bases using the angle, ϕ , of the vector $p - c$. If we let (u, v) be the coordinates of $p - c$ then $\phi(u, v) = \text{atan2}(v, u)^1$, and

$$r = (\cos \phi, \sin \phi), \quad t = (-\sin \phi, \cos \phi). \quad (2)$$

Let V be the 2×2 matrix that transforms a Cartesian gradient, $g = (g_x, g_y)$, into a radial gradient, $\gamma = (\gamma_r, \gamma_t)$, such that

$$\gamma = Vg = R_{-\phi}g = \begin{bmatrix} r^T \\ t^T \end{bmatrix} g = \begin{bmatrix} r^T g \\ t^T g \end{bmatrix}. \quad (3)$$

Now assume that the patch has been rotated about its center by some angle, θ . This yields a new local coordinate system and gradient, given by

$$r' = R_\theta r, \quad t' = R_\theta t, \quad g' = R_\theta g. \quad (4)$$

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

$$\begin{aligned} \gamma' &= (g'^T r', g'^T t') \\ &= ((R_\theta g)^T R_\theta r, (R_\theta g)^T R_\theta t) \\ &= (g^T R_\theta^T R_\theta r, g^T R_\theta^T R_\theta t) \\ &= (g^T r, g^T t) \\ &= \gamma. \end{aligned} \quad (5)$$

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. We can see that this is not the case for Cartesian gradients because $g' = R_\theta g \neq g$, except when θ is a multiple of 2π or $g = 0$.

B. Approximate Radial Gradient Transform

The RGT allows us to create gradient histograms directly from the image grid, rather than rotating the pixels. However,

¹ $\text{atan2}(y, x) = 2 \tan^{-1} \frac{y}{\sqrt{x^2 + y^2} + x}$ is the angle of the vector (x, y) .

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 alleviate this problem, we propose approximating the exact basis vector, r and t , by new vectors, \hat{r} and \hat{t} , such that we can compute an Approximate Radial Gradient Transform (ARGT) directly from image pixels.

As with the RGT, these basis vectors are all functions of the position in the patch, (u, v) , relative to the center, where $(u, v) = p - c$. We define the quantized angle, $\hat{\phi}(u, v)$, for a point in a patch as

$$\hat{\phi}(u, v) = \frac{2\pi}{N} \left\lfloor \frac{N}{2\pi} \phi(u, v) + \frac{1}{2} \right\rfloor \bmod 2\pi \quad (6)$$

where N is the number of quantized directions. Fig. 2 shows an example comparison between the exact (r) and approximate (\hat{r}) bases for choice of $N = 8$. We thus define \hat{r} and \hat{t} as

$$\hat{r} = (\cos \hat{\phi}, \sin \hat{\phi}), \quad \hat{t} = (-\sin \hat{\phi}, \cos \hat{\phi}) \quad (7)$$

which lead to an approximate transform matrix, \hat{V} .

$$\hat{V} = \begin{bmatrix} \hat{r}^T \\ \hat{t}^T \end{bmatrix} = \begin{bmatrix} \cos \hat{\phi} & \sin \hat{\phi} \\ -\sin \hat{\phi} & \cos \hat{\phi} \end{bmatrix} = R_{-\hat{\phi}} \quad (8)$$

We can then compute the approximate radial gradient as

$$\hat{\gamma} = \hat{V}g. \quad (9)$$

We demonstrate the rotation-invariance of the ARGV in Fig. 3. Given two matching circular patches, we compute a histogram of Cartesian and approximate radial gradients. For both gradient types, the histograms are similar for the database and query patch. When the query patch is rotated by 90° , the Cartesian gradient histogram also rotates by 90° . However, the approximate radial gradient histogram maintains nearly the same shape. The variation between the upright and rotated query histograms is due to the basis vector approximation. In the next sections we analyze this variation in more depth.

For further analysis, we wish to relate the exact and approximate gradients. To do so, we express $\hat{\gamma}$ in terms of γ by applying Eqs. 3 and 9

$$\hat{\gamma} = \hat{V}g = \hat{V}V^{-1}\gamma = \hat{V}V^T\gamma \quad (10)$$

$$= R_{-\hat{\phi}}R_\phi\gamma = R_{\phi-\hat{\phi}}\gamma = R_\theta\gamma \quad (11)$$

where we let $\theta = \phi - \hat{\phi}$. The approximate radial gradient is simply the exact gradient rotated by θ . We next consider the relationship between quantized versions of γ and $\hat{\gamma}$.

III. GRADIENT QUANTIZATION ERROR

To form a histogram-of-gradient feature descriptor, we wish to quantize gradients to reduce data and improve robustness. We have shown that the exact RGT is invariant to rotation, and we now consider the difference between quantizing exact and approximate radial gradients. In particular, we want to know the probability of error, \bar{P}_E , of quantizing $\hat{\gamma}$ differently than γ . We define this measure as

$$\bar{P}_E = \int_{\gamma \in \mathbb{R}^2} P_E(\gamma) f_\Gamma(\gamma) d\gamma \quad (12)$$

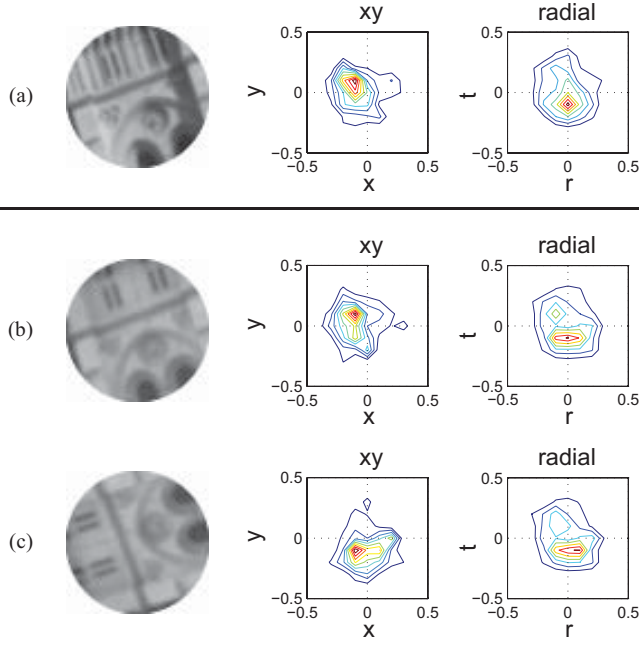


Fig. 3. (a) Database. (b) Upright Query. (c) Rotated Query. Illustration of rotation invariance with the ARG. A database patch (top left) produces a gradient histogram in both Cartesian (top middle) and radial (top right) domains. We extract similar histograms from an upright query patch (center left) and its rotated version (bottom left). 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.

where $P_E(\gamma)$ is the conditional probability of error given the exact γ , and $f_\Gamma(\gamma)$ is the probability distribution of exact radial gradients. To compute \tilde{P}_E , we need an expression for each component of the integrand. For the gradient distribution we fit a model to empirical data. We define the conditional probability of error as

$$P_E(\gamma) = \Pr(Q(\gamma; q) \neq Q(\hat{\gamma}; q)) \quad (13)$$

where $Q(\gamma; q)$ is the gradient quantizer function, and q is the scale parameter of the quantizer. To compute this probability we must choose a quantizer, and use the conditional pdf and cdf of $\hat{\gamma}$ given γ . Before doing so, we first model the radial gradient distribution.

A. Radial Gradient Probability Model

To model the distribution of radial gradients, we first note their relationship to Cartesian gradients. Since the transform matrix of the RGT, V , is orthonormal, the magnitude distribution of γ is the same as that of g . Because we only consider circular patches, the angle of rotation of the RGT, ϕ , is uniform over the range $[0, 2\pi)$. Therefore, if we assume that the distribution of g is isotropic, then the orientation distribution for γ will also be isotropic. We can make this assumption because we do not orient our patches along any dominant gradient. Thus, we model the pdf of γ as a function of only magnitude. To do so, we use the Laplace mixture model proposed by Levin and Weiss [24]

$$f_\Gamma(\gamma) = \frac{w}{2s_1} \exp\left(-\frac{\|\gamma\|}{s_1}\right) + \frac{1-w}{2s_2} \exp\left(-\frac{\|\gamma\|}{s_2}\right) \quad (14)$$

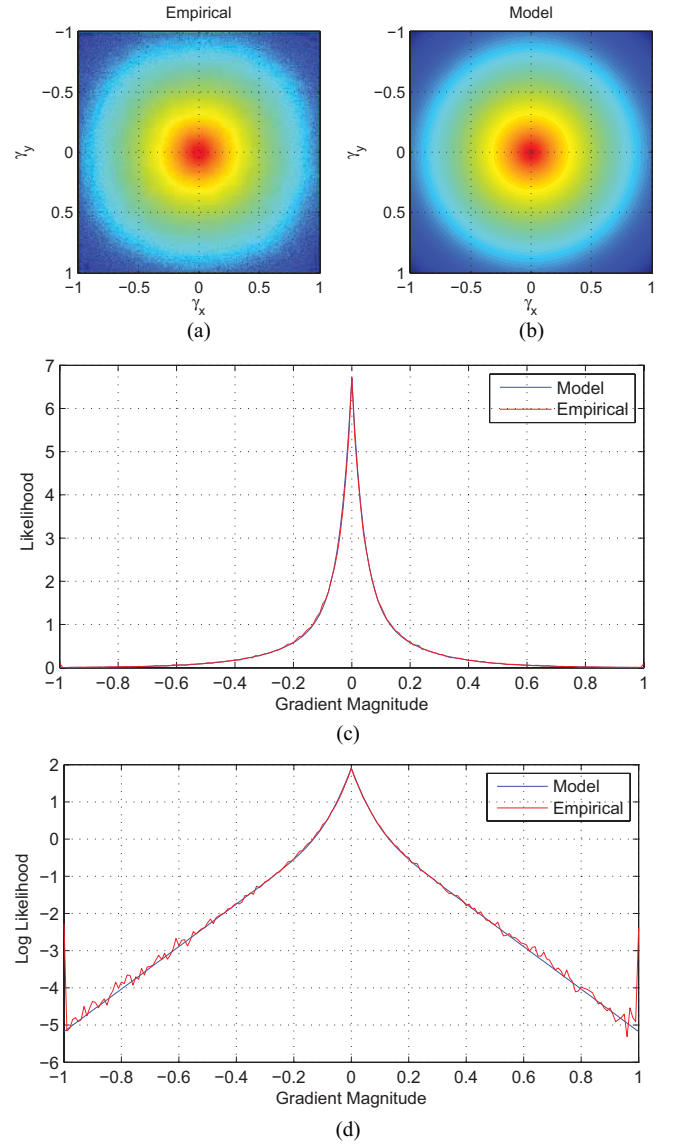


Fig. 4. Mixture of Laplacian model for radial gradients. (a) We show the joint log-likelihood as an image for empirical data. (b) Corresponding model. (c) We also show a 1-D slice through the x -axis for linear likelihood. (d) Log likelihood.

where the weight, w , is 0.6, and the scale parameters, s_1 and s_2 , are 0.175 and 0.04, respectively. We estimated these parameters using patches from the dataset provided by Winder *et al.* [25]. To better represent the patches on which we compute radial gradients, we randomly orient the patches, and normalize their intensities. Fig. 4 demonstrates that this model accurately captures the distribution. The model essentially assumes that gradients are bimodal, either coming from textured or flat regions. Within each of these types of regions, the gradients follow a Laplace distribution.

B. Gradient Quantizers

We consider two different 9-bin quantizers; a scalar quantizer, and a polar vector quantizer, each of which is illustrated in Fig. 5. Let Q_{scalar} be the scalar quantizer, and Q_{polar} be the

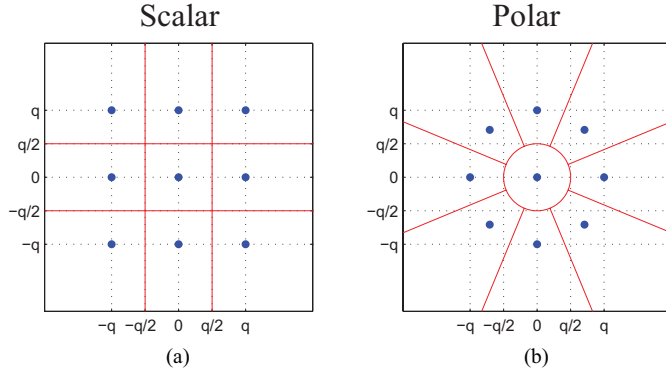


Fig. 5. Gradient quantizers relative to the quantization step size, q . (a) We consider a scalar. (b) Polar vector quantizer. The blue dots indicate the representative values, and the red lines mark the quantization cell boundaries.

polar quantizer. We define the scalar quantizer as

$$Q_{\text{scalar}}(\gamma; q) = (h(\gamma_r; q), h(\gamma_i; q)) \quad (15)$$

where $h(\cdot)$ is a three-level 1D quantizer given by

$$h(x; q) = \begin{cases} -q & \text{if } x < -\frac{q}{2} \\ q & \text{if } x > \frac{q}{2} \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

We define the polar quantizer as

$$Q_{\text{polar}}(\gamma; q) = \begin{cases} (0, 0) & \text{if } \|\gamma\| < \frac{q}{2} \\ (q \cos \beta, q \sin \beta) & \text{otherwise} \end{cases} \quad (17)$$

where $\beta = \frac{\pi}{M} \lfloor \frac{M}{\pi} \psi + \frac{1}{2} \rfloor$, $M = 8$ is the number of gradient bins (excluding the central bin), and ψ is the angle of γ .

C. Conditional PDF

We now consider the conditional pdf of $\hat{\gamma}$ given γ . Let Θ and Φ be random variables associated with θ and ϕ , respectively. Because points in a circular patch are evenly distributed over all angles we may assume that $\Phi \sim U(0, 2\pi)$. Recall that θ is the negative quantization error of ϕ due to the ARG. Thus, $\Theta \sim U(-\frac{\pi}{N}, \frac{\pi}{N})$ because ϕ is uniformly distributed, and quantized with a step-size of $2\pi/N$. Let ψ be the angle of γ , and $\hat{\psi}$ be the angle of $\hat{\gamma}$

$$\psi = \text{atan2}(\gamma_i, \gamma_r), \quad \hat{\psi} = \text{atan2}(\hat{\gamma}_i, \hat{\gamma}_r). \quad (18)$$

Then the angle of $\hat{\gamma}$ is $\hat{\psi} = \psi + \theta$. Therefore, the distribution of the angle of $\hat{\gamma}$ given γ is

$$\hat{\Psi}|\psi \sim U\left(\psi - \frac{\pi}{N}, \psi + \frac{\pi}{N}\right). \quad (19)$$

We let $\hat{\Gamma}$ be the random variable associated with γ such that

$$\hat{\Gamma} = \|\gamma\| \begin{bmatrix} \cos(\hat{\Psi}) \\ \sin(\hat{\Psi}) \end{bmatrix} = \|\gamma\| \begin{bmatrix} \cos(\psi + \Theta) \\ \sin(\psi + \Theta) \end{bmatrix} \quad (20)$$

The conditional pdf of $\hat{\Gamma}$ given γ is an arc of a circle with a radius of $\|\gamma\|$

$$f_{\hat{\Gamma}|\Gamma}(\hat{\gamma}|\gamma) = \frac{N}{2\pi\gamma} \delta(\|\hat{\gamma}\| - \|\gamma\|) \left[\sigma\left(\hat{\psi} - \psi + \frac{\pi}{N}\right) - \sigma\left(\hat{\psi} - \psi - \frac{\pi}{N}\right) \right]. \quad (21)$$

where $\delta(\cdot)$ is the Dirac delta function, $\sigma(\cdot)$ is the step function.

Given the conditional pdf, we can integrate it to find the corresponding conditional cdf $F_{\hat{\Gamma}|\Gamma}(x, y)$

$$F_{\hat{\Gamma}|\Gamma}(x, y) = \int_{-\infty}^x \int_{-\infty}^y f_{\hat{\Gamma}|\Gamma}\left(\begin{bmatrix} u \\ v \end{bmatrix}\right) du dv. \quad (22)$$

We describe the details of this integration in Appendix I.

D. Scalar Quantization Error

Now that we have obtained an expression for the conditional cdf, we can formulate the probability of error. Consider the distance between the gradient and the decision boundaries. The boundaries are $\pm q/2$ and $\pm\infty$ for both gradient components. Let B be the set of quantization boundaries

$$B = \{-\infty, -q/2, q/2, \infty\}. \quad (23)$$

Choose u^+ , u^- , and v^+ , v^- as the upper and lower bounds of the cell in which γ falls

$$u^+ = \min_i B_i, \quad B_i > \gamma_x \quad (24)$$

$$u^- = \max_i B_i, \quad B_i \leq \gamma_x \quad (25)$$

$$v^+ = \min_i B_i, \quad B_i > \gamma_y \quad (26)$$

$$v^- = \max_i B_i, \quad B_i \leq \gamma_y. \quad (27)$$

We define the probability of error, $P_E(\gamma)$, as the likelihood that $\hat{\gamma}$ is outside of the quantization cell that contains γ

$$P_E(\gamma) = \Pr(Q(\gamma; q) \neq Q(\hat{\gamma}; q)) \quad (28)$$

$$= 1 - F_{\hat{\Gamma}|\Gamma}(u^+, v^+) - F_{\hat{\Gamma}|\Gamma}(u^-, v^-) + F_{\hat{\Gamma}|\Gamma}(u^-, v^+) + F_{\hat{\Gamma}|\Gamma}(u^+, v^-). \quad (29)$$

Using this expression and the gradient model, we can compute the integrand of Eq. 12. Fig. 6 shows the conditional probabilities and the weighted conditional probabilities. Note that the maximum conditional probability of error is 100%. This peak error occurs at the cell boundary intersections, and at the intersection of the axes and cell boundaries.

E. Polar Quantizer Error

We now analyze the ARG. quantization error for the polar quantizer. Analysis of this quantizer is much simpler than the scalar quantizer because the conditional pdf of $\hat{\gamma}$ is also polar. Let $C = \{\frac{2\pi}{M}i \mid i = 0 \dots M\}$ be the set of angles at the center of each bin (excluding the center bin), and c^* be the angle closest to ψ

$$c^* = \underset{c \in C}{\text{argmin}} |c - \psi|. \quad (30)$$

The conditional probability of error, $P_E(\gamma)$, is given by

$$P_E(\gamma) = \begin{cases} 0 & \text{if } \|\gamma\| < \frac{q}{2} \\ 1 - \alpha(c_\gamma, w_\gamma, c^*, \pi/M) & \text{otherwise} \end{cases} \quad (31)$$

where $\alpha(\cdot)$ is defined in Eq. 67. This expression captures the amount of overlap between the angle intervals of $f_{\hat{\Gamma}|\Gamma}(\hat{\gamma})$ and the bin into which γ falls. Because of the circular shape of the central bin, if $\|\gamma\| < q/2$ then $\hat{\gamma}$ is guaranteed to be quantized

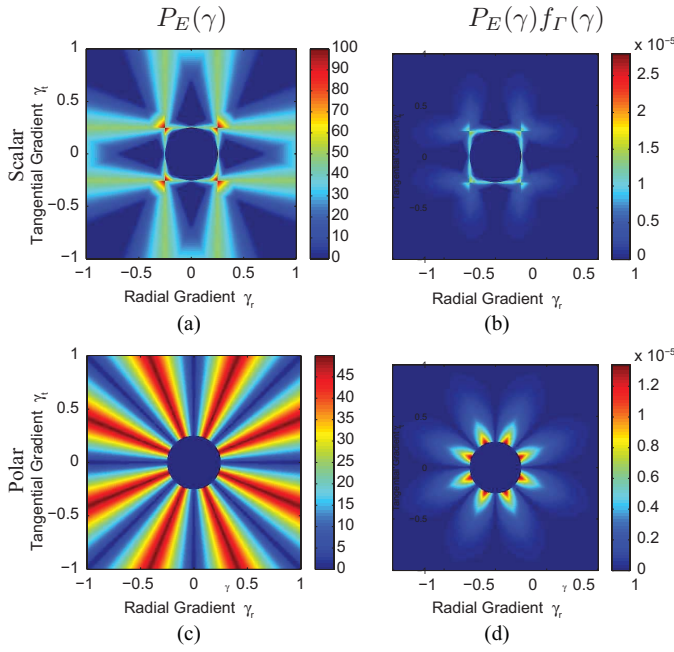


Fig. 6. (a) Conditional probability of error, and (b) weighted probability of error for (c) a scalar and (d) polar quantizer with scale parameter of 0.5.

to the same bin. Furthermore, we can see that for $N \geq M$, the maximum conditional error will be 50%, and occur when γ falls on the decision boundary. This is half of the maximum error for the scalar quantizer.

F. Quantization Error Comparison

Having developed an ARGV quantization error model for both quantizers, we wish to compare their results as a function of the step size, q , and the number of ARGV directions, N . Additionally, we verify our models against experimental results. To obtain our experimental data, we use patches from Winder *et al.* [25]. Additionally, we randomly orient and intensity normalize the patches. We compute and quantize the exact radial gradient for each pixel in each patch. We then do the same for the approximate radial gradients. Given these two sets of quantized values, we simply measure the frequency with which they differ. We repeat this experiment for a range of $N = \{4, 6, 8, 10, 12, 14, 16\}$ and q between 0 and 1.5.

Fig. 7 shows the quantization error rate versus step size for empirical and analytical data using a scalar quantizer. The error decreases as we increase the number of ARGV directions or the quantization step size. We observe that our model behaves very similarly to the experimental results. The greatest discrepancy occurs around $q = 1$, and is due to slight differences between the gradient distribution model and the true distribution. For very small values of q the error is small because the on-axis bins are vanishingly small. The likelihood of γ falling into the central bins shrinks faster than the corresponding increase in error. For large values of q the error approaches zero because all gradients fall into the same large bin. This leads to a peak in the error at $q \approx 0.1$.

Fig. 8 shows similar results for the polar quantizer. Again, the experimental and theoretical curves align well. The pri-

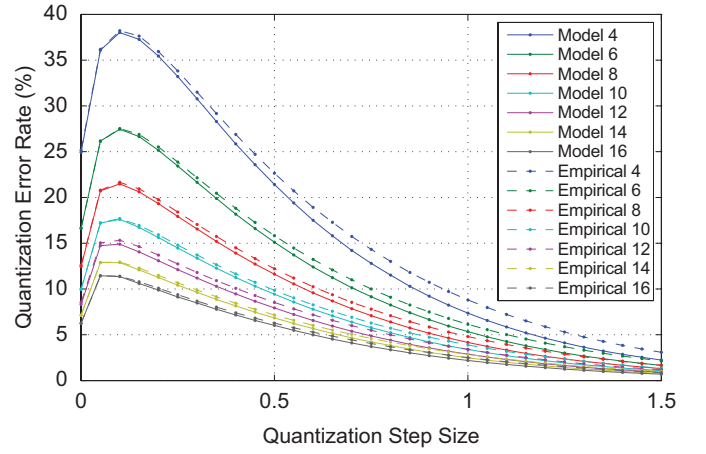


Fig. 7. Quantization error rate versus step size for empirical and analytical data using a scalar quantizer. The model for N ARGV directions (solid) matches the empirical data for N directions (dashed). The error decreases as the number of ARGV directions or the quantization step size increases.

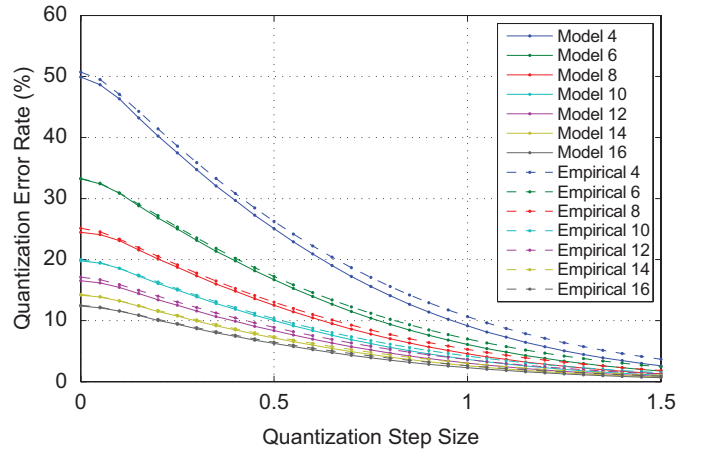


Fig. 8. Quantization error rate versus step size for empirical and analytical data using a polar quantizer. The model for N ARGV directions (solid) matches the empirical data for N directions (dashed). The error decreases as the number of ARGV directions or the quantization step size increases.

mary difference between the polar and scalar quantizers is that the error decreases monotonically with increasing q for the polar quantizer. This is because the error for the central bin of the polar quantizer is always zero. The error of the other bins is constant in q . Therefore, as q decreases, the zero-error region shrinks and is replaced by non-zero error.

G. Histogram Error

The probability of misquantization exaggerates the effect on the radial gradient histogram because not all misquantizations occur in the same bin. To cause an error in the histogram, a gradient must erroneously leave a bin and not be replaced by a misquantization from another bin. We now consider the effect that the ARGV has on histograms of quantized gradients. Let $H = [H_1, \dots, H_M]$ be a histogram of quantized gradients, and $\mathcal{B} = \{\mathcal{B}_1, \dots, \mathcal{B}_M\}$ be the associated quantization bins. The histogram error, E_i , in bin \mathcal{B}_i is

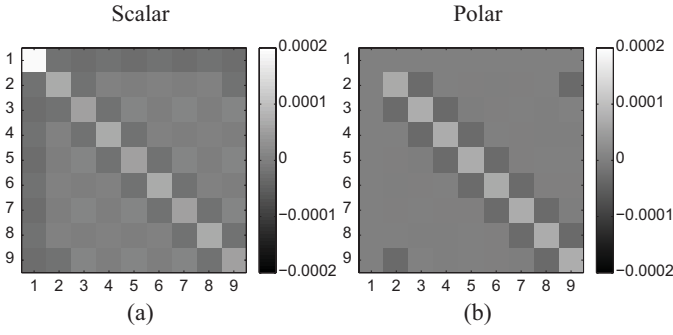


Fig. 9. Covariance matrix of the histogram error for (a) scalar and (b) polar quantizers. The first dimension is the central bin, followed by the outer bins in counter-clockwise order. The errors in a bin are negatively correlated with those in neighboring bins.

given by

$$E_i = \sum_{\mathbf{x} \in P} \mathbf{1}(\hat{\gamma}(\mathbf{x}) \in B_i \mid \gamma(\mathbf{x}) \notin B_i) - \sum_{\mathbf{x} \in P} \mathbf{1}(\hat{\gamma}(\mathbf{x}) \notin B_i \mid \gamma(\mathbf{x}) \in B_i) \quad (32)$$

where \mathbf{x} is the position in a patch, P . The error vector is thus $E = [E_1, \dots, E_M]$. So that we may compare histograms with different numbers of samples, we consider the probability mass function, $\hat{H} = \frac{1}{|P|}H$, and the associated error vector, $\bar{E} = \frac{1}{|P|}E$, where $|P|$ is the number of gradients in the histogram.

We empirically measure the statistics of the histogram error vector, \bar{E} , on the same *Liberty* patch dataset that we used previously. First, we examine the covariance matrix of \bar{E} for both quantizers, as shown in Fig. 9. The first dimension of the error corresponds to the central bin. The covariance for this dimension is zero for the polar quantizer, as expected from the misquantization analysis. For the scalar quantizer, the central bin is the primary source of errors.

For $N = 8$ ARGT directions, the angle of $\hat{\gamma}$ cannot differ from that of γ by more than 22.5° . For the polar quantizer, this means that errors in neighboring bins are negatively correlated, but errors in non-neighboring bins are independent. For the scalar quantizer, the bins do not subtend a constant angle, and large magnitude gradients can be misquantized to non-neighboring bins. This effect is apparent in the elements of the covariance matrices that are off of the tri-diagonal.

In Fig. 10 we plot the distribution of the error magnitude, $\|\bar{E}\|_2$, for both quantizers with a range of step sizes. We have chosen the range of 0.2 to 0.6 for the step sizes because it is the optimal range of both quantizers. (The optimal parameters are selected in Section V. The optimal step size is 0.5 for the scalar quantizer, and 0.3 for the polar quantizer.) In Fig. 11, we compare the mean values of each distribution shown in Fig. 10. The histogram error between the exact and approximate RGT decreases as the step size increases. The polar quantizer has a lower error for a given step size. At their optimal step sizes, the histogram error is 15% higher for the polar quantizer than the scalar quantizer.

Note that the mean error for both quantizers is less than about 0.03, or three percent of \bar{H} . In Fig. 12, we show the effect that the ARGT has on the discriminatory power

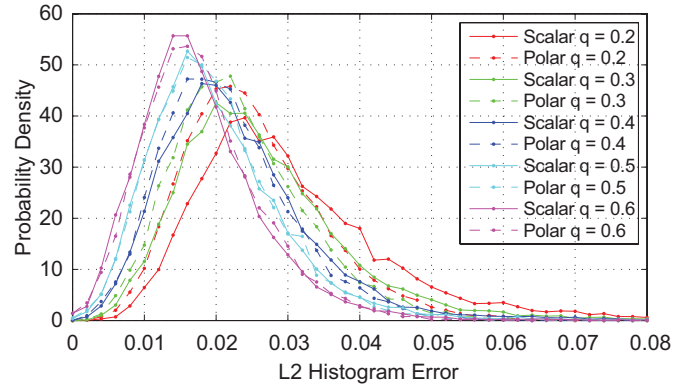


Fig. 10. Distribution of L_2 histogram errors between the exact and approximate RGT for polar and scalar quantizers. The error decreases as the quantization step size, q , increases.

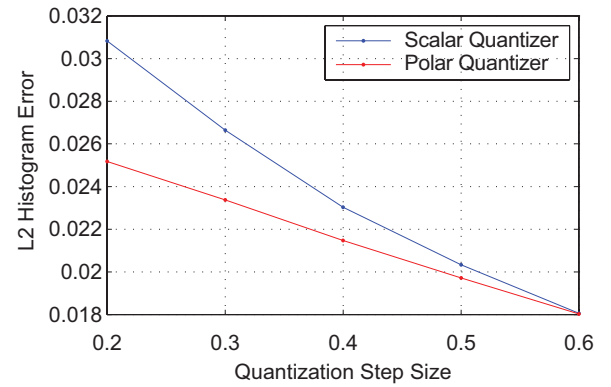


Fig. 11. Mean of histogram errors between the exact and approximate RGT versus quantization step size for polar and scalar quantizers.

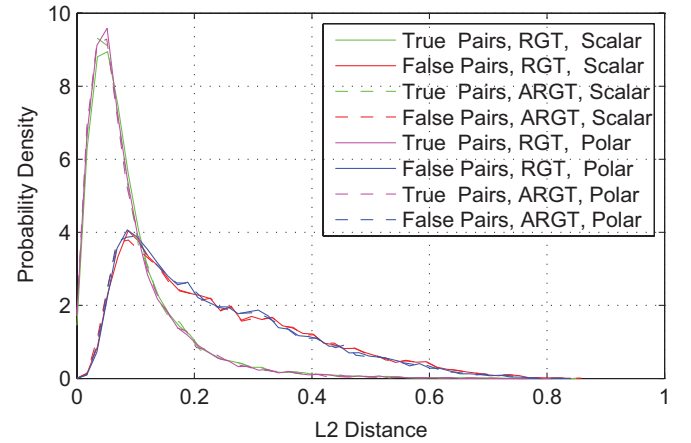


Fig. 12. Histogram of distances between matching patches and non-matching patches for exact and approximate radial gradients using the polar quantizer. The ARGT has very little effect on the histograms. Both quantizers have a step size of 0.4.

of the histogram of gradients. Using the *Liberty* data set, we plot the distribution of L_2 -distances between histograms of gradients from pairs of matching patches, and similarly for non-matching pairs. We do so for both scalar and polar quantizers with the same step size. Note that the distributions are very similar for both quantizers, and that the ARGT has

little effect on the distributions. The ARG T is therefore not detrimental to the histogram of quantized gradients. In the next section, we will show how to apply the ARG T to construct robust, rotation-invariant, local feature descriptors.

IV. ROTATION-INVARIANT FAST FEATURES

In this section, we build a local feature algorithm based on histograms of quantized approximate radial gradients. We first briefly describe our interest point detection algorithm, and then explain how we produce a descriptor with the ARG T.

A. Feature Detection

We now describe our interest point detection algorithm. To achieve a balance of speed and robustness we use a Difference of Boxes (DoB) filter kernel to detect blobs.

1) *Difference of Boxes*: We take the filters proposed by Agrawal *et al.* [26] as a starting point. The filter response is the simple weighted difference of two box filters that are centered on the same point but have different scales. For a scale parameter, s , the inner box has width $2s + 1$ and the outer box is roughly twice the size with width $4s + 1$. The filter response, $F(x, y, s)$, is thus given by

$$F(x, y, s) = B(x, y, s) - B(x, y, 2s) \quad (33)$$

where $B(x, y, s)$ is the area-normalized box filter function

$$B(x, y, s) = (2s + 1)^{-2} \Sigma(x, y, s). \quad (34)$$

We define $\Sigma(x, y, s)$ as the sum of pixel values within a box of width $2s + 1$, centered at (x, y) . We compute these sums efficiently by using an integral image. If we let $I(x, y)$ be input image, and $S(x, y)$ be the associated integral image, then

$$S(x, y) = \sum_{v=0}^y \sum_{u=0}^x I(u, v). \quad (35)$$

$$\Sigma(x, y, s) = S(x+s, y+s) + S(x-s-1, y-s-1) - S(x+s, y-s-1) - S(x-s-1, y+s). \quad (36)$$

With this method we can compute a filter response at any scale or position from a single integral image.

2) *Scale-Space*: To be robust to image scale changes, we compute filter responses for a range of filter scales, yielding a stack of filtered images. For reasonable coverage of possible scales, we choose a range that covers ~ 3 octaves. To do so, we let s cover all integers $1 \dots 8$. Agrawal *et al.* [26] propose computing a full, dense scale-space. To reduce the computational complexity, we propose using a *pyramid* scale space, where each scale layer is down-sampled by a factor that matches the filter scale. In our scheme, the first scale is computed on the full resolution, and the subsequent scales are down-sampled by factors of $2\times$, $3\times$, $4\times$, *etc.*

To compute a descriptor from a given location in scale-space, anti-aliased pixels values must be computed at the correct scale. Instead of recomputing these values with the integral image, or via a mipmap with trilinear interpolation, we simply reuse our DoB filter results. To do so, we store the inner box-filter values, $B(x, y, s)$, in a separate image scale-space memory buffer.

3) *Local Scale-Space Extrema*: As with other interest point detectors (*e.g.* SIFT, SURF and CenSurE), we use local extrema to find repeatable points in scale space. To find the neighbors of a pixel at position (x, y, s) , we first consider the 8 neighbors within the same scale, given by $\{(x \pm s, y \pm s, s), (x, y \pm s, s), (x \pm s, y, s)\}$. We then find the nearest existing pixels in the scales above and below, $(x_+, y_+, s + 1)$ and $(x_-, y_-, s - 1)$, where

$$x_- = (s - 1) \lfloor x/(s - 1) + 0.5 \rfloor \quad (37)$$

$$x_+ = (s + 1) \lfloor x/(s + 1) + 0.5 \rfloor \quad (38)$$

and similarly for y_+ and y_- .

Given these central pixels above and below, we find their 8 neighbors as before. We consider (x, y, s) to be an interest point if its filter response is positive and greater than its neighbors', or negative and less than its neighbors'. We call this the *inter-scale* detection scheme. For speed, we also propose a simpler *intra-scale* scheme wherein we consider a point to be a local extremum if it is maximal or minimal relative to its 8 neighbors on the same scale. While the *inter* scheme provides full scale-space localization, the *intra* scheme describes points at multiple salient scales, and is faster. Additionally, it has been demonstrated by Wu *et al.* [27] that multi-scale description can be beneficial. The results of both schemes are shown later in Section VI.

As a final filtering stage we remove edge responses by computing a Harris corner score around each detected interest point. We only keep local maxima that have a positive Harris corner score.

B. Feature Description

Given the scale space and interest points described in the previous section, we now describe how to apply the ARG T to produce robust, fast feature descriptors that are invariant to image rotation.

1) *Patch Extraction*: During interest point detection, we have computed an image scale-space that corresponds to the filter response scale-space. As a result, we have anti-aliased pixel values at the scale and location of each interest point. For an interest point at coordinates (x, y, s) , we extract a circular local patch with diameter p that is centered at (x, y) in scale level s . Because of the way we store the scale-space, the diameter of the patch in the scale space is sp . Pixel coordinates which belong to the patch are in the set P

$$P(s) = \left\{ (u, v) \mid \sqrt{(su - x)^2 + (sv - y)^2} \leq \frac{sp}{2} \right\}. \quad (39)$$

2) *Orientation Assignment*: We assign a canonical orientation for the patch by finding the dominant Cartesian-gradient orientation within the patch. For speed, we use a look-up table to convert from Cartesian to polar coordinates. This is possible because there are only 512 possible values for pixel differences in an 8-bit grayscale image. This yields a 512×512 table. Using the table, we quickly compute the magnitude and orientation of each gradient and form a magnitude-weighted angle histogram. We use 72 orientation bins (5° resolution), to which we apply a 3-bin average filter to smooth the histogram.

Given the smoothed histogram, we compute a robust orientation. Rather than producing multiple orientations, like SIFT, we choose a single, repeatable angle. If the second most dominant bin is less than 90% of the maximal bin then we choose the dominant bin's angle. If the two most dominant bins are within 10% of each other then we choose the canonical orientation as the angle that is to the right of the angle that bisects the two most dominant bins.

3) *Intensity Normalization*: To be invariant to image contrast changes we normalize the gradients by the standard deviation, σ , of the patch's intensity values. This normalization is applied during the radial gradient quantization. We compute $\sigma = \sqrt{v - \mu^2}$, where μ is the mean pixel value and v is the mean squared pixel value. To avoid traversing the patch for a second time, we compute these values concurrently with the orientation assignment.

4) *Radial Gradient Quantization*: We compute radial gradients at each pixel using the ARGV. The ARGV must incorporate proper baseline normalization because diagonal pixel neighbors are farther than parallel pixel neighbors. Let b be the distance between two pixels in the ARGV, and q be the desired gradient quantizer step-size. We combine the quantizer parameter, intensity and baseline normalization by multiplying pixel differences by $\alpha = bq\sigma$. Finally, we obtain quantized radial gradients by rounding to each component to $\{-1, 0, 1\}$, yielding one of nine possible gradients.

We perform this operation efficiently by using a look-up table that has the same dimensions as the patch. Let $(u, v) \in P$ be pixel coordinates in the local patch centered at (x, y, s) , and N be the number of ARGV directions. We define i_r to be the radial ARGV angle bin index for a pixel, and i_t to be the tangential bin index such that

$$i_r(u, v) = \left\lfloor \frac{N}{2\pi} \phi(u, v) + \frac{1}{2} \right\rfloor \bmod N \quad (40)$$

$$i_t(u, v) = \left\lfloor \frac{N}{2\pi} \left(\phi(u, v) + \frac{\pi}{2} \right) + \frac{1}{2} \right\rfloor \bmod N \quad (41)$$

where $\phi(u, v) = \text{atan2}(v - y, u - x)$. We choose pixel offsets for the radial direction as (δ_x^r, δ_y^r) , and similarly for the tangential direction as (δ_x^t, δ_y^t) . For the choice of $N = 8$ directions, we choose our offsets and scaling as below

$$(\delta_x, \delta_y, \alpha) = \begin{cases} (s, 0, q\sigma) & \text{if } i = 0 \\ (s, s, \sqrt{2}q\sigma) & \text{if } i = 1 \\ (0, s, q\sigma) & \text{if } i = 2 \\ (-s, s, \sqrt{2}q\sigma) & \text{if } i = 3 \\ (-s, 0, q\sigma) & \text{if } i = 4 \\ (-s, -s, \sqrt{2}q\sigma) & \text{if } i = 5 \\ (0, -s, q\sigma) & \text{if } i = 6 \\ (s, -s, \sqrt{2}q\sigma) & \text{if } i = 7. \end{cases} \quad (42)$$

Note that there are only two possible values for α . We need only modify a single number in our look-up table per patch because q is constant and σ is constant over the patch. Similarly, the δ are constant within a scale-space level and need only be computed once for each scale during program initialization. We thus compute the approximate radial

gradient, \hat{g} , as

$$\hat{g}(u, v) = \alpha^{-1} \begin{bmatrix} B(u + \delta_x^r, v + \delta_y^r, s) - B(u, v, s) \\ B(u + \delta_x^t, v + \delta_y^t, s) - B(u, v, s) \end{bmatrix}. \quad (43)$$

where $B(u, v, s)$ is the box filtered scale-space. Having efficiently computed \hat{g} , we quantize it using $Q_{\text{scalar}}(\cdot)$ as defined in Eq. 15.

5) *Spatial Binning*: The next stage of our proposed descriptor algorithm is spatial binning, wherein we form histograms of quantized radial gradients. Each histogram comes from a spatially contiguous sub-region of the local patch. As shown by Chandrasekhar *et al.* [28], Winder *et al.* [25], and Mikolajczyk *et al.* [13], a polar spatial binning arrangement performs better than a Cartesian arrangement such as those used in SIFT [11] and SURF [12]. We therefore adopt a polar binning scheme. For rotation-invariance, we rotate the spatial bins to align with the feature orientation. We do so efficiently with a lookup table that contains one spatial bin configuration for each of the 72 orientations and 8 scales.

6) *Descriptor Computation*: To form the descriptor, \mathcal{D} , we loop over each pixel location in the patch and form a 9-bin gradient histogram, H , for each spatial bin

$$\mathcal{D} = [H_1 \dots H_{n_b}]. \quad (44)$$

Let the set of pixels in the k^{th} spatial bin be P_k , and H_k^i be the i^{th} gradient bin of the k^{th} spatial bin. We assume that the quantizer, $Q(\cdot)$, returns integers in the range of 1 to 9

$$H_k^i = \frac{1}{|P_k|} \sum_{(u,v) \in P_k} \mathbf{1}(Q(\hat{g}(u, v)) = i). \quad (45)$$

We can then optionally compress the descriptor using the method described in [28]. This scheme uses a lattice quantizer that preserves the L_1 -norm of the histograms which comprise the descriptor. The results of such compression are shown in Section VI.

V. GRADIENT QUANTIZER OPTIMIZATION

Our descriptor algorithm has several parameters which can be varied to tune performance. In this section we will optimize the choice of gradient quantizer. We refer the reader to [29] for optimal values of the other parameters. We use the Receiver Operating Characteristic (ROC) to measure the quality of a descriptor. To capture the ROC performance of a descriptor algorithm with a single number we use the Area Under the Curve (AUC). To measure the ROC, we use ten thousand patches provided by Winder *et al.* [25]. Note that radial gradients are not required if the patch is already oriented, as is the case with the Liberty patches. Therefore, we randomly orient one patch from each pair to test the robustness of gradient quantization. We then re-orient each patch with the previously described orientation assignment.

Fig. 13 shows the ROC performance versus quantizer step size. Both quantizers peak with similar ROC performance, though the scalar quantizer performs slightly better than the polar quantizer. The peak value occurs for different step sizes of 0.4 for the scalar, and 0.25 for the polar quantizer. The ROC score of the scalar quantizer drops more slowly from

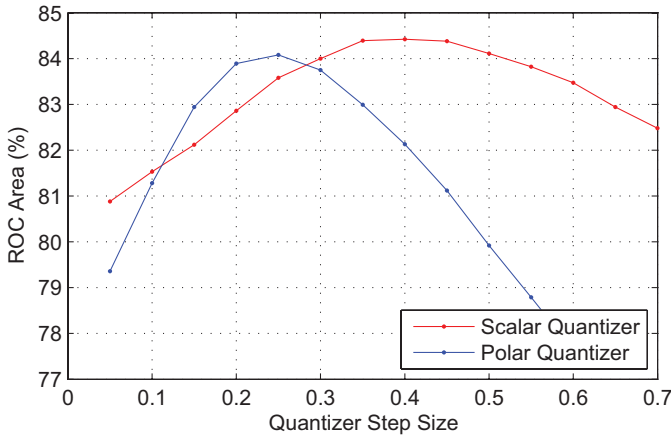


Fig. 13. ROC results for scalar and polar quantizers versus quantization step size. The scalar quantizer slightly outperforms the polar quantizer.

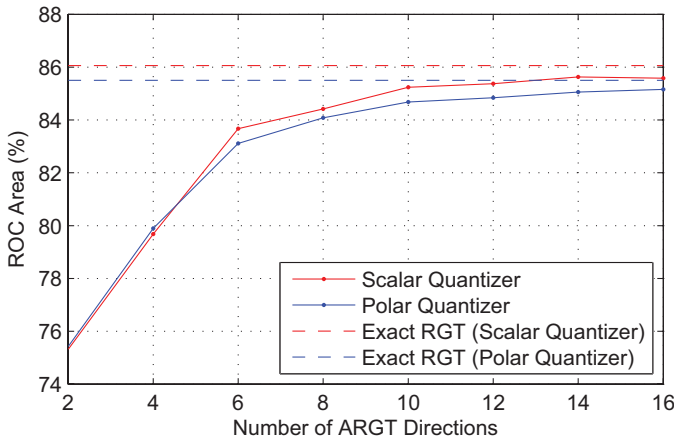


Fig. 14. ROC results for scalar and polar quantizers versus the number of ARG T directions. The performance approaches that of the exact RGT as the number of directions increases, and drops steeply below six directions.

its peak value than that of the polar quantizer. Therefore, the scalar quantizer is less sensitive to the precise choice of step size than is the polar quantizer. The performance of the polar quantizer drops off more quickly from its peak value than does the scalar quantizer. Thus, we prefer the scalar quantizer because it is faster and performs better.

Using the optimal quantizer step size, we vary the number of ARG T directions from 2 to 16. Fig. 14 shows the resulting ROC performance. We also compare against the performance of the exact RGT. The performance slowly degrades as the number of directions drops from 16 to 6. Below 6 directions the performance drops quickly. As expected, the ARG T approaches the performance of the exact RGT as the number of directions increases. The behavior is very similar for both scalar and polar quantizers. We use 8 ARG T directions because that is smallest number of directions that performs well and is easily computed directly on the image raster.

VI. IMAGE MATCHING AND RETRIEVAL

In this section, we evaluate the performance of RIFF at the image level. To do so we consider three measures;

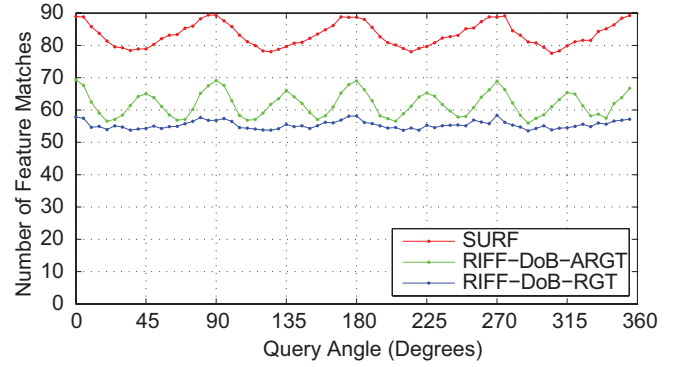


Fig. 15. Orientation invariance of pairwise image matching. Curves are averaged over 100 image pairs of CD covers. ETH-SURF and exact RGT RIFF oscillate with a 90° period due to box filtering. ARG T RIFF oscillates with a 45° period due to the eight-direction ARG T.

execution profile, pairwise orientation invariance, and retrieval accuracy.

A. Execution Profile

First, we compare the execution profiles of RIFF and SURF on an Intel Xeon 2.8 GHz. To represent a typical feature extraction task, we extract 500 features from a 512×512 -pixel image. The number of features is set by tuning the interest point detection threshold for each algorithm. To time the SURF algorithm we use the number provided by the ETH SURF implementation, averaged over ten runs. SURF takes 250 milliseconds and RIFF takes 15 milliseconds. RIFF is thus $16.5\times$ faster than SURF.

We also measure the portion of time that each algorithm spends in sub-tasks. SURF spends 26% of the time detecting interest points, 41% of the time assigning orientations to interest points, and 33% of the time computing descriptors. RIFF spends 36% of the time computing DoB responses, 11% of the time finding scale-space extrema, and 53% of the time computing descriptors. Note that SURF spends the largest portion of time assigning orientations. RIFF is faster because it uses a less complex scale-space, reuses the scale-space for descriptor computation, and does not interpolate pixels or rotate gradients.

B. Rotation Invariance

We use pairwise image matching to ensure that the overall RIFF algorithm is invariant to rotation. For each of 100 pairs of CD covers, we rotate one image in five degree increments and measure the number of feature matches. The image pairs come from the MPEG CDVS dataset [30]. For feature matching we use a ratio-test [31] followed by geometric consistency checking with an affine RANSAC. To eliminate edge effects due to image rotation, we use a 480-pixel diameter circular cropping of the images and add a 100-pixel border to all sides.

In Fig. 15, we compare SURF to RIFF with and without the RGT approximation. SURF oscillates with a period of 90 degrees due to box filtering used in both interest point detection and descriptor computation. Similarly, RIFF with the exact RGT oscillates with a 90-degree period but with

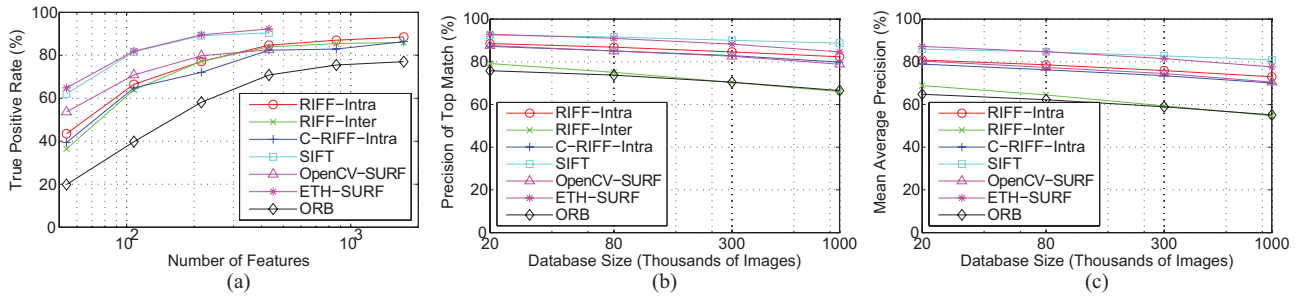


Fig. 16. (a) Pairwise (TPR). (b) Retrieval (PAO). (c) Retrieval (MAP). Recognition results for three variants of the proposed RIFF algorithm (Intra, Inter, Compressed Intra), as well as SIFT and two SURF implementations. The left column is True Positive Rate (TPR) for pairwise matching subject to a false positive constraint of 1%. The center column is the Precision At One (PAO) measure for retrieval, and the right column is the mean average precision (MAP). RIFF-Intra performs on par with OpenCV SURF and SIFT, while ETH SURF outperforms all other algorithms.

much smaller variation. It only employs box filtering in the interest point detection, but not in descriptor computation. The ARGV causes an oscillation with a 45 degree period due to the 8-direction approximation. Interestingly, the ARGV improves the number of feature matches. This improvement is likely because the ARGV has a soft-binning effect on the gradient quantization. The magnitude of the oscillations in RIFF-ARGV are comparable to those in SURF, and we conclude that the algorithm is sufficiently rotation-invariant.

C. Retrieval

We demonstrate RIFF's retrieval performance on a challenging, large-scale, real-world dataset. We also compare RIFF with other state-of-the-art algorithms. To do so, we use the MPEG CDVS dataset and evaluation framework [30]. This dataset contains five classes of images; text/graphics, museum paintings, video frames, common objects, and buildings/landmarks. There are 2500, 455, 500, 14935, and 10200, images in each of the respective classes. In the first three classes, there exists a clean reference image.

Using this set of images, MPEG defines a pairwise matching experiment with 16 k matching pairs, and 172 k non-matching pairs of images. We report the true positive rate for each class, subject to the constraint that the overall false positive rate is less than 1%. For retrieval, MPEG defines a database of 18 thousand images and 11 thousand query images. An additional 1 million distractor images from Flickr are added to the database images. Using these images, we report the precision of the first match, or Precision At One (PAO), as well as the Mean Average Precision (MAP) for each query class. To demonstrate scalability, we report recognition as a function of the number of database images.

We use a state-of-the-art image retrieval system that uses a vocabulary tree with and weak geometric re-ranking, followed pairwise matching and strong geometric consistency checking. Our system is similar to the ones used by Girod *et al.* [32], and Philbin *et al.* [9]. For pairwise matching we use a ratio test as proposed by Lowe [11], followed by an affine RANSAC.

Fig. 16 shows the results of these experiments for SIFT [33], OpenCV SURF [34], ETH SURF [35], ORB [16], RIFF-Intra, RIFF-Inter and Compressed-RIFF (~100 bits/feature). We observe that RIFF-Intra significantly outperforms RIFF-Inter in all categories. This is likely because the feature

budget is spent on the most salient features. Compression only causes a small drop in RIFF's performance. Approximately $5\times$ as many features are required for RIFF-Intra to match the performance of SURF at pairwise matching. However, because RIFF is $16\times$ faster than SURF per feature, the RIFF can match SURF's performance with a $3\times$ the speed. For retrieval, we use the same number of features (~800) for all algorithms. RIFF-Intra performs on par with SURF when the database has 1 million images. ORB performs significantly worse than the other algorithms.

VII. CONCLUSION

We have shown the RGT to be a practical method for producing rotation-invariant histograms of gradients. Our computationally efficient approximation of the exact RGT enables computation directly from the pixel values, leading to a very fast algorithm. Our analysis of the ARGV quantifies the effects of the approximation on gradient quantization. We have shown that the approximation does not significantly degrade the rotation-invariance property of the RGT, and that, in fact, it improves pairwise image matching.

Our analysis of gradient quantization has produced several interesting conclusions. First, we have shown that an isotropic mixture-of-Laplacians is an excellent fit for the distribution of radial gradients that occur in image patches around interest points. Using this gradient model, we accurately modelled the quantization error introduced by the ARGV. Using this quantization error model, in addition to empirical ROC measurements, we have shown that a low-complexity scalar quantizer can outperform a more complex polar vector quantizer for histogram-of-gradients feature descriptors. We have also evaluated ROC performance as a function of the number of ARGV directions. The results of this analysis have shown that fewer than 8 directions significantly degrades performance, while more than 8 directions only slightly improves performance. We have therefore chosen the 8-direction approximation because it is easily computed from the image raster.

We have also described how to use the ARGV in a complete, local feature descriptor algorithm, and shown that the resulting descriptors are competitive with state-of-the-art descriptors for retrieval. The RIFF descriptor pipeline is $16\times$ faster than SURF. This high speed enables web-scale image processing with modest hardware. For example, RIFF can extract 500

features from one million 500×500 -pixel images in ~ 4 hours on a single core of an Intel Xeon 2.8 GHz. The same task would take almost three days with SURF. RIFF can process an even larger, web-scale database of one billion images in ~ 11 days on a single 16-core server. At these high speeds, feature extraction is no longer the bottle-neck of image processing.

APPENDIX I

APPROXIMATE RADIAL GRADIENT CDF

In this appendix we compute the conditional cdf of an approximate radial gradient, $\hat{\gamma}$, given the exact radial gradient, γ . In Section III-C, we computed the conditional pdf, $f_{\hat{\gamma}| \gamma}(\hat{\gamma}|\gamma)$. Given this pdf, we now find the corresponding conditional cdf, $F_{\hat{\gamma}| \gamma}(x, y)$

$$F_{\hat{\gamma}| \gamma}(x, y) = \int_{-\infty}^x \int_{-\infty}^y f_{\hat{\gamma}| \gamma} \left(\begin{bmatrix} u \\ v \end{bmatrix} \right) du dv. \quad (46)$$

The cdf is the fraction of the pdf that is less than the coordinates (x, y) . To compute this integral we take advantage of the pdf's geometry. The pdf is non-zero on an arc of radius $\|\gamma\|$, that subtends $2\pi/N$ radians. Since we are only interested in the fraction of this arc, we normalize the coordinates such that the arc lies on the unit circle.

To find the cdf, we first find the portions of the unit circle that are left and below a point (x, y) . These portions are defined by a set of angles, S_θ

$$S_\theta = \{\theta \mid \cos(\theta) < x, \sin(\theta) < y\}. \quad (47)$$

The conditional pdf can likewise be represented as a set of angles, S_γ

$$S_\gamma = \{\theta \mid \frac{\pi}{N} > |\theta - \psi|\}. \quad (48)$$

Both of these sets can be represented as angle intervals. We represent the intervals by their centers, c , and half-widths, w , such that the corresponding interval is $[c - w, c + w]$. The interval for S_γ is given by

$$c_\gamma = \psi, \quad w_\gamma = \frac{\pi}{N}. \quad (49)$$

The set S_θ may contain 0, 1, or 2 intervals, depending on x and y . These intervals are defined by sets of centers, C , and widths, W . We first define six regions of the plane, $R_1 \dots R_6$. These regions are shown in Fig. 17 and defined below

$$R_1 = \{x, y \mid 1 < x, 1 < y\} \quad (50)$$

$$R_{2A} = \{x, y \mid x < -1\} \quad (51)$$

$$R_{2B} = \{x, y \mid y < -1\} \quad (52)$$

$$R_{2C} = \{x, y \mid 1 < x^2 + y^2, x < 0, y < 0\} \quad (53)$$

$$R_2 = R_{2A} \cup R_{2B} \cup R_{2C} \quad (54)$$

$$R_{3A} = \{x, y \mid |x| \leq 1, 1 < y\} \quad (55)$$

$$R_{3B} = \{x, y \mid 1 < x^2 + y^2, -1 \leq x \leq 0, 0 \leq y\} \quad (56)$$

$$R_3 = R_{3A} \cup R_{3B} \quad (57)$$

$$R_{4A} = \{x, y \mid |y| \leq 1, 1 < x\} \quad (58)$$

$$R_{4B} = \{x, y \mid 1 < x^2 + y^2, -1 \leq y \leq 0, 0 \leq x\} \quad (59)$$

$$R_4 = R_{4A} \cup R_{4B} \quad (60)$$

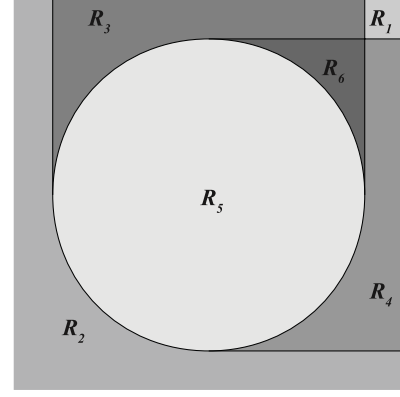


Fig. 17. Regions for the conditional cdf of the gradient quantization error.

$$R_5 = \{x, y \mid x^2 + y^2 \leq 1\} \quad (61)$$

$$R_6 = \{x, y \mid 1 < x^2 + y^2, 0 < x < 1, 0 < y < 1\}. \quad (62)$$

We can then get the appropriate set of arcs for the region into which (x, y) falls

$$C(x, y) = \begin{cases} \{0\} & \text{if } (x, y) \in R_1 \\ \{\} & \text{if } (x, y) \in R_2 \\ \{\pi\} & \text{if } (x, y) \in R_3 \\ \{\frac{3\pi}{2}\} & \text{if } (x, y) \in R_4 \\ \{\frac{1}{2}(3\pi - \sin^{-1} y - \cos^{-1} x)\} & \text{if } (x, y) \in R_5 \\ \left\{ \frac{1}{2}(\sin^{-1} y + \cos^{-1} x) \right\} & \text{if } (x, y) \in R_6. \end{cases} \quad (63)$$

$$W(x, y) = \begin{cases} \{2\pi\} & \text{if } (x, y) \in R_1 \\ \{\} & \text{if } (x, y) \in R_2 \\ \{\pi - \cos^{-1} x\} & \text{if } (x, y) \in R_3 \\ \{\frac{\pi}{2} + \sin^{-1} y\} & \text{if } (x, y) \in R_4 \\ \{\frac{1}{2}(\pi + \sin^{-1} y - \cos^{-1} x)\} & \text{if } (x, y) \in R_5 \\ \left\{ \frac{1}{2}(\sin^{-1} y - \cos^{-1} x) \right\} & \text{if } (x, y) \in R_6. \end{cases} \quad (64)$$

From these arcs we define the cdf

$$F_{\hat{\gamma}| \gamma}(x, y) = \frac{1}{2w_\gamma} \sum_{i=1}^{|C|} \alpha(c_\gamma, w_\gamma, C_i(u, v), W_i(u, v)). \quad (65)$$

where $(u, v) = (x, y)/\|\gamma\|$, and α is the overlap of two arcs

$$\alpha(c_1, w_1, c_2, w_2) = \min(\alpha_1, \alpha_2) \quad (66)$$

$$\alpha_1 = 2 \min(w_1, w_2) \quad (67)$$

$$\alpha_2 = \sum_{i=1}^2 \max(0, w_1 + w_2 - d_i). \quad (68)$$

where d is the set of two distances between arc centers

$$d = \{|c_2 - c_1|, 2\pi - |c_2 - c_1|\}. \quad (69)$$

REFERENCES

- [1] G. Takacs, M. E. Choubassi, Y. Wu, and I. Kozintzev, "3D mobile augmented reality in urban scenes," in *Proc. IEEE Conf. Multimedia Expo.*, Sep. 2011, pp. 1-4.
- [2] D. M. Chen, S. S. Tsai, R. Grzeszczuk, R. Vedantham, and B. Girod, "Streaming mobile augmented reality on mobile phones," in *Proc. Int. Symp. Mixed Augm. Real.*, Oct. 2009, pp. 1-2.

- [3] L. Ballan, M. Bertini, A. Del Bimbo, and A. Jain, "Automatic trademark detection and recognition in sport videos," in *Proc. IEEE Int. Conf. Multimed. Expo.*, vol. 23, Apr. 2008, pp. 901–904.
- [4] T. Chattopadhyay and A. Sinha, "Recognition of trademarks from sports videos for channel hyperlinking in consumer end," in *Proc. IEEE 13th Int. Symp., Consum. Electron.*, May 2009, pp. 943–947.
- [5] S. Arafat, S. Husain, I. Niaz, and M. Saleem, "Logo detection and recognition in video stream," in *Proc. 5th Int. Conf. Dig. Inf. Manag.*, Jul. 2010, pp. 163–168.
- [6] N. M. Cheung, D. M. Chen, V. Chandrasekhar, R. Vedantham, S. S. Tsai, G. Takacs, R. Grzeszczuk, and B. Girod, "Video retrieval from a single query frame," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 1–6.
- [7] D.-G. Sim, H.-K. Kim, and D.-I. Oh, "Translation, scale, and rotation invariant texture descriptor for texture-based image retrieval," in *Proc. Int. Conf. Image Process.*, 2000, pp. 742–745.
- [8] S. S. Tsai, D. M. Chen, G. Takacs, V. Chandrasekhar, R. Vedantham, R. Grzeszczuk, and B. Girod, "Fast geometric re-ranking for image based retrieval," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 1–4.
- [9] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization - improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [10] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 5, pp. 530–535, May 1997.
- [11] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [12] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.
- [13] 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.
- [14] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, Y. Reznik, R. Grzeszczuk, and B. Girod, "Compressed histogram of gradients: A low-bitrate descriptor," *Int. J. Comput. Vis.*, vol. 1, pp. 1–16, May 2011.
- [15] M. Ambai and Y. Yoshida, "CARD: Compact and real-time descriptors," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 97–104.
- [16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
- [17] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2548–2555.
- [18] N. Kingsbury, "Rotation-invariant local feature matching with complex wavelets," in *Proc. Eur. Conf. Signal Process.*, 2006, pp. 1–5.
- [19] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2005, pp. 1–8.
- [20] V. Chandrasekhar, G. Takacs, and D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "CHoG: Compressed histogram of gradients, a low bitrate descriptor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2009, pp. 1–16.
- [21] P. Brasnett and M. Z. Bober, "Robust visual identifier using the trace transform," in *Proc. IET Vis. Inf. Eng. Conf.*, Jul. 2007, pp. 1–8.
- [22] C. Roover, C. Vleeschouwer, F. Lefebvre, and B. Macq, "Robust video hashing based on radial projections of key frames," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 4020–4037, Oct. 2005.
- [23] J. Seo, J. Haitsma, T. Kalker, and C. Yoo, "A robust image fingerprinting system using the radon transform," in *Proc. Signal Process. Image Commun. Conf.*, 2004, pp. 1–15.
- [24] A. Levin and Y. Weiss, "User assisted separation of reflections from a single image using a sparsity prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 1647–1654, May 2007.
- [25] S. Winder, G. Hua, and M. Brown, "Picking the best DAISY," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2009, pp. 1–8.
- [26] M. Agrawal, K. Konolige, and M. Blas, "CenSurE: Center surround extremas for realtime feature detection and matching," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2008, pp. 102–115.
- [27] Z. Wu, Q. Ke, J. Sun, and H.-Y. Shum, "A multisample, multitree approach to bag-of-words image representation for image retrieval," in *Proc. Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1–6.
- [28] V. Chandrasekhar, Y. Reznik, G. Takacs, D. M. Chen, S. S. Tsai, R. Grzeszczuk, and B. Girod, "Study of quantization schemes for low bitrate CHoG descriptors," in *Proc. IEEE Int. Workshop Mobile Vis.*, Jun. 2010, pp. 1–6.
- [29] G. Takacs, B. Girod, L. Guibas, and R. Grzeszczuk, "Unified tracking and recognition with rotation-invariant fast features," Ph.D. thesis, Univ. Stanford, Stanford, CA, 2012.
- [30] Y. Reznik, G. Cordara, and M. Bober, *Evaluation Framework for Compact Descriptors for Visual Search*. San Francisco, CA: Academic, 2011.
- [31] M. Brown and D. Lowe, "Unsupervised 3D object recognition and reconstruction in unordered datasets," in *Proc. 5th Int. Conf. 3-D Imag. Modell.*, Jun. 2005, pp. 56–63.
- [32] B. Girod, V. Chandrasekhar, D. M. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham, "Mobile visual search," *IEEE Signal Process. Mag.*, vol. 28, no. 7, pp. 61–76, Jul. 2011.
- [33] A. Vedaldi. "UCLA Vision Library's SIFT Code." (2009) [Online]. Available: <http://www.vlfeat.org/>
- [34] G. Bradski, "The OpenCV Library," *Dr. Dobbs' Journal of Software Tools*. (2000) [Online]. Available: <http://opencv.willowgarage.com/wiki/>
- [35] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, May 2006, pp. 1–6.



Gabriel Takacs received the B.S. degree in engineering from Harvey Mudd College, Claremont, CA, in 2005, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 2007 and 2012, respectively.

He was a Senior Researcher with Nokia Research Center, Palo Alto, CA. He is currently a Scientist with Microsoft Corporation, Sunnyvale, CA, where he is involved in research and development of augmented reality and image-based retrieval.



Vijay Chandrasekhar received the B.S. and M.S. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2005. He is currently pursuing the Ph.D. degree with the IVMS Group, Stanford University, Stanford, CA, under the supervision of Prof. Girod.

He is an Active Participant with the ongoing MPEG standard: Compact Descriptors for Visual Search. His current research interests include low-bitrate image retrieval.



Sam S. Tsai received the B.S. and M.S. degrees in electronics engineering and electronics from National Chiao Tung University, Hsinchu, Taiwan, in 2001 and 2003, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Stanford University, Stanford, CA.

He was with Realtek Semiconductor Corp., Hsinchu, Taiwan, from 2003 to 2007, where he was a Systems Design Engineer and was involved in developing digital multimedia systems. His current research interests include use and compression of geometric information and natural scene text information for mobile visual search.



David Chen received the B.S. and M.S. degrees in electrical engineering from Stanford University, Stanford, CA, where he is currently pursuing the Doctorate degree.

He is currently involved in research on the design of mobile visual search systems that support low-latency queries against large visual databases of products or objects. Many of the algorithms developed have been successfully incorporated in real-time mobile systems for recognizing media covers, book spines, outdoor landmarks, and TV content. In

2011, he was engaged in research with the Bicoastal Team at Stanford and Columbia to develop a novel personalized television news system. His current research interests include computer vision, machine learning, multimedia analysis, and image and video compression.



Radek Grzeszczuk received the M.S. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, in 1994 and 1998, respectively.

He is currently a Principal Scientist with Microsoft Corporation, Sunnyvale, CA, where he is leading a team in researching image search, augmented reality, and large-scale urban 3-D reconstruction. He was a Principal Researcher Lead with Nokia Research Center, Palo Alto, CA, where he joined in 2006 as a Principal Researcher. He was a Senior Researcher with the Architecture Research Laboratory, Intel

Corp., Santa Clara, CA, USA, where he was involved in research on image-based rendering, parallel algorithms, and performance analysis of large-scale applications in image processing, physical simulation, optimization, and machine learning. He has authored or co-authored over 80 conference and journal papers.



Bernd Girod received the M.S. degree from the Georgia Institute of Technology, Atlanta, and the Doctorate degree in engineering from the University of Hannover, Hannover, Germany.

He has been with the Information Systems Laboratory, Stanford University, Stanford, CA, since 1999, where he is currently a Professor of electrical engineering and computer science, and the Senior Associate Dean of Online Learning and Professional Development with the School of Engineering. He was a Professor with the Electrical Engineering

Department, University of Erlangen-Nuremberg, Erlangen, Germany. He has authored or co-authored over 500 conference and journal papers and six books. His current research interests include networked media systems.

Dr. Girod was the recipient of the EURASIP Signal Processing Best Paper Award in 2002, the IEEE Multimedia Communication Best Paper Award in 2007, the EURASIP Image Communication Best Paper Award in 2008, the EURASIP Signal Processing Most Cited Paper Award in 2008, the EURASIP Technical Achievement Award in 2004, and the Technical Achievement Award from the IEEE Signal Processing Society in 2011. He is an entrepreneur who is involved in several startup ventures, among them are Polycom, Vivo Software, 8×8 , and RealNetworks. He is a EURASIP Fellow and a member of the German National Academy of Sciences (Leopoldina).