Quantization Schemes for Low Bitrate Compressed Histogram of Gradients Descriptors

Vijay Chandrasekhar* Yuriy Reznik‡ Gabriel Takacs* David Chen* Sam Tsai*
Radek Grzeszczuk† Bernd Girod*

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

† Nokia Research Center
Palo Alto, CA
radek.grzeszczuk@nokia.com

‡ Qualcomm Inc.
San Diego, CA
yreznik@qualcomm.com

Abstract

We study different quantization schemes for the Compressed Histogram of Gradients (CHoG) image feature descriptor. We propose a scheme for compressing distributions called Type Coding, which offers lower complexity and higher compression efficiency compared to tree-based quantization schemes proposed in prior work. We construct optimal Entropy Constrained Vector Quantization (ECVQ) code-books and show that Type Coding comes close to achieving optimal performance. The proposed descriptors are 16× smaller than SIFT and perform on par. We implement the descriptor in a mobile image retrieval system and for a database of 1 million CD, DVD and book covers, we achieve 96% retrieval accuracy using only 4 kilobytes of data per query image.

1. Introduction

Mobile phones have evolved into powerful image and video processing devices, equipped with high-resolution camera, color displays, and hardware-accelerated graphics. This enables a class of applications which use the camera phone to initiate search queries about objects in visual proximity to the user. Such applications can be used for identifying products, comparison shopping, finding information about movies, CDs, real estate or products of the visual arts. For these applications, a query photo is taken by a mobile device and compared against a database on a remote server. The size of the data sent over the network needs to be as small as possible to reduce latency and improve user experience. In this work, we study descriptor compression techniques and show that compressed descriptors can reduce query latency significantly in mobile image retrieval systems.

1.1. Prior Work

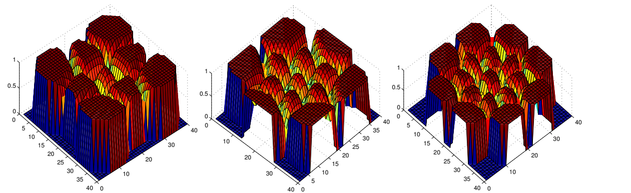
Low bitrate descriptors are of increasing interest in the computer vision community. Often, feature vectors are reduced in size by decreasing the dimensionality of descriptors via Principle Component Analysis (PCA) or Linear Discriminant Analysis (LDA) [13, 11]. In [4, 21], we study dimensionality reduction and entropy coding of SIFT and SURF descriptors. Winder *et al.* [23] combine the use of PCA with additional optimization of gradient and spatial binning parameters as part of the training step. The disadvantages of PCA and LDA approaches is high computational complexity, and the risk of overtraining for descriptors from a particular data set. Further, with PCA and LDA, descriptors cannot be compared in the compressed domain if entropy coding is employed. Yeo *et al.* [24] reduce the bitrate of descriptors by using random projections on SIFT descriptors to build binary hashes. Shakhnarovich, in his thesis [19], uses a machine learning technique called Similarity Sensitive Coding to train binary codes on image patches. However, hashing schemes do not perform well at low bitrates [5]. In [5], we propose construction of low-bitrate feature descriptors by using Compressed Histogram of Gradients (CHoG). CHoG descriptors can be compared directly in the compressed domain eliminating the need for decompression in the descriptor matching process.

1.2. Contributions

In this work, we use the framework of CHoG [5] and study alternate techniques that can be used for quantization and compression of descriptor data. Our contributions in this paper are as follows:

- We propose a new scheme for compressing distributions called Type Coding. Type Coding provides a better trade-off in bitrate versus Equal Error Rate (EER) compared to tree based quantization schemes employed by the original CHoG design [5]. Compared to Huffman-

978-1-4244-7030-3/10/$26.00 ©2010 IEEE



(*DAISY-9*) (*DAISY-13*) (*DAISY-17*)

Figure 1. The DAISY spatial binning configurations used for $n = 9, 13, 17$ spatial bins.

tree coding, we obtain 20% to 50% decrease in bitrate at a given EER. We show that descriptors compressed with Type Coding scheme can be compared in the compressed domain. We compute the Entropy Constrained Vector Quantization (ECVQ) performance bound for descriptor's data and show that Type Coding comes close to achieving it.

- We compare our low bitrate descriptors with several other schemes from the literature, and show that that our proposed descriptors outperform them.
- Finally, we evaluate the performance of CHoG descriptors in a mobile image retrieval system. For a database of 1 million CD, DVD and book covers, we achieve 96% retrieval accuracy using only 4 kilobytes of data per query image. Similar retrieval accuracy with SIFT would require 16× as much data, and with compressed JPEG images, would require 10× as much data.

1.3. Paper Outline

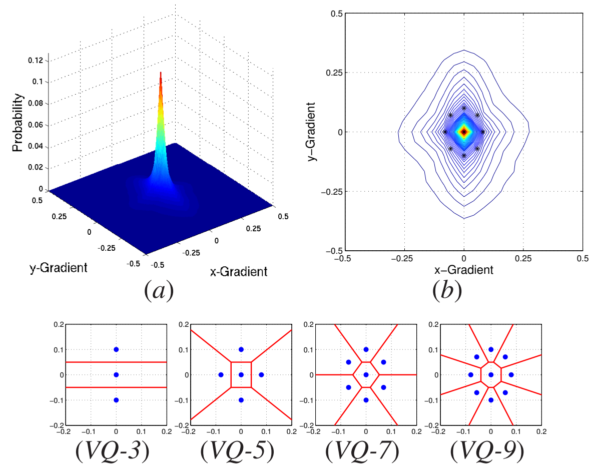
In Section 2, we review the Histogram-of-Gradients (HoG) descriptor used in our work. In Section 3, we discuss three different schemes for compressing distributions, and compare their performance. Finally, in Section 4, we evaluate performance of compressed descriptors in a mobile image retrieval system.

2. Descriptor Design

A number of different feature descriptors are based on the distribution of gradients within a patch of pixels. Lowe [14], Bay *et al.* [2], Dalal and Triggs [10], Winder *et al.* [23], as well as current authors *et al.* [5] have proposed histogram of gradient based descriptors. Our present design is based on the CHoG descriptor [5].

2.1. Computing Histograms of Gradients

We start with a canonical patch extracted around an interest point at the detected scale and orientation. We normalize the pixel values of each patch and compute local image gradients d_x and d_y. The patch is divided into localized cells based on the the DAISY configurations proposed in [22, 23]. We use overlapping regions for spatial binning which improves the performance of the descriptor by making it more robust to interest point localization error. The soft assignment is made such that each pixel contributes to multiple spatial bins with normalized Gaussian weights that sum to 1.



(a) (b)

(*VQ-3*) (*VQ-5*) (*VQ-7*) (*VQ-9*)

Figure 2. The joint (d_x, d_y) gradient distribution (a) over a large number of cells, and (b), its contour plot. The greater variance in y-axis results from aligning the patches along the most dominant gradient after interest point detection. The quantization bin constellations VQ-3, VQ-5, VQ-7 and VQ-9 and their associated veronoi cells are shown at the bottom.

Next, we quantize the gradient histogram in each spatial bin. Let $P_{D_x,D_y}(d_x, d_y)$ be the normalized joint (x,y)-gradient histogram in each spatial bin. We coarsely quantize the 2D gradient histogram and capture the histogram directly into the descriptor. We approximate $P_{D_x,D_y}(d_x, d_y)$ as $\hat{P}_{\hat{D}_x,\hat{D}_y}(\hat{d}_x, \hat{d}_y)$ for $(\hat{d}_x, \hat{d}_y) \in S$, where S represents a small number of quantization centroids or bins as shown in Figure 2. Based on underlying gradient statistics, we perform a Vector Quantization (VQ) of the gradient distribution into a small set of bin centers, S, shown in Figure 2. We call these gradient bin configurations VQ-3, VQ-5, VQ-7 and VQ-9. Similar to soft spatial binning, we assign each (d_x, d_y) pair to multiple bin centers with normalized Gaussian weights. As we increase the number of bin centers, we obtain a more accurate approximation of the gradient distribution. Before we study different compression schemes, we briefly discuss the evaluation procedure used.

2.2. Descriptor Performance Evaluation

For evaluating the performance of low bitrate descriptors, we use the two data sets provided by Winder and Brown in their most recent work [23], *Notre Dame* and *Liberty*. For algorithms that require training, we use use the *Notre Dame* data set, while we perform our testing on the *Liberty* set. We use the methodology proposed in Winder and Brown [23] for evaluating descriptors. We compute symmetric Kullback-Leibler (KL) distance between each matching and non-matching pair of descriptors. From these distances, we obtain a Receiver Operating Characteristic (ROC) curve which plots correct match fraction against incorrect match fraction. We compare our low bitrate descriptors to the SIFT descriptor.We focus on descriptors that perform on par with SIFT and are in the range of 50-100 bits.

3. Descriptor Compression

Our goal is to produce low bitrate CHoG descriptors while maintaining the highest possible fidelity. In this Section, we discuss three different schemes for compressing histograms: Huffman Coding, Type Coding and ECVQ. For each scheme, we quantize the gradient histogram in each cell individually and map it to an index. The indices are then encoded with fixed-length or entropy codes, and the bitstream is concatenated together to form the final descriptor. We also experimented with joint coding of the gradient histograms in different cells, but this did not yield any practical gain.

Let m represent the number of gradient bins. Let $P = [p_1, p_2, ...p_m] \in \mathbb{R}^m_+$ be the original distribution as described by histogram, and $Q = [q_1, q_2, ...q_m] \in \mathbb{R}^m_+$ be the quantized probability distribution defined over the same sample space.

There are several measures that can be used for determining the degree of mismatch between distributions [9],

- L_α-norms over a vector of probability differences:

$$||P-Q||_\alpha = \left(\sum_i |p_i - q_i|^\alpha\right)^{1/\alpha}, \quad \alpha \geqslant 1,$$

- KL (Kullback-Leibler) divergence:

$$D(P||Q) = \sum_i p_i \log_2 \frac{p_i}{q_i},$$

- Symmetric KL divergence:

$$J(P,Q) = D(P||Q) + D(Q||P).$$

These measures are related [9], for example,

$$\tfrac{1}{2\ln 2}||P-Q||_1^2 \leqslant D(P||Q) \leqslant \tfrac{1}{\ln 2}||P-Q||_\infty^2 \sum_i \tfrac{1}{q_i}.$$

As mentioned earlier, we are primarily interested in the symmetric KL distance.

3.1. Huffman Tree Coding

Given a probability distribution, one to way to compress it is to construct and store a Huffman tree for this distribution [5]. The reconstructed distribution $Q = Q(\ell_1, \ldots, \ell_m)$ becomes

$$q_i = 2^{-\ell_i}, \quad \ell_i \in \mathbb{Z}_+, \quad \sum_i 2^{-\ell_i} = 1 \qquad (1)$$

where $\ell_1, \ldots, \ell_m$ denote the lengths of Huffman code words (paths from root to leaves in the Huffman tree).

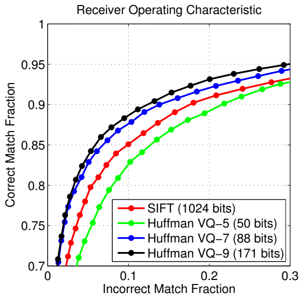


Figure 3. ROC curves for compressing distributions with Huffman scheme for the DAISY-9 configuration for the *Liberty* data set. The CHoG descriptor at 88 bits outperforms SIFT at 1024 bits.

The number of Huffman trees $T(m)$ utilized by such a scheme can be estimated by considering labeling of all possible rooted binary trees with m leaves

$$T(m) < m!\, C_{m-1},$$

where $C_n = \frac{1}{n+1}\binom{2n}{n}$ is a Catalan number. Hence, the index of a Huffman tree representing distribution P with fixed-length encoding requires at most

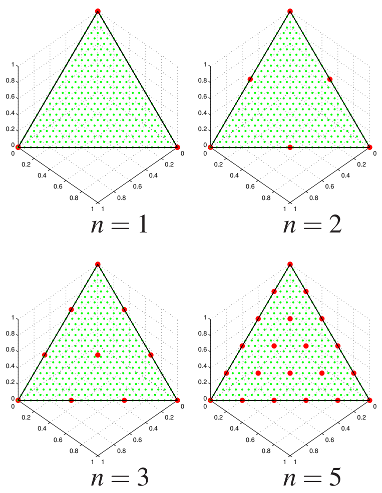
$$R_{\mathrm{Huf}}(m) \leqslant \lceil \log_2 T(m) \rceil \sim m \log_2 m + O(m). \qquad (2)$$

bits to encode.

Implementation. Quantization is implemented by a standard Huffman tree construction algorithm, requiring $O(m \log m)$ operations, where m is the number of bins in the gradient histogram. All unique Huffman trees are enumerated and their indices are stored in memory. The number of Huffman trees for $m = 3,5,7,9$ are $3, 75, 4347$ and 441675 respectively. The number of trees grows very rapidly with m and tree enumeration becomes impractical beyond $m = 9$. We implemented both fixed-length and entropy coding of tree indices. For $m \leqslant 7$, we found entropy coding to be useful, resulting in savings of approximately $10 - 20\%$ in the bitrate. This compression is achieved by using context adaptive binary arithmetic coding.

Example. Let $m = 5$ corresponding to the VQ-5 gradient bin configuration. Let $P = [0.1, 0.3, 0.2, 0.25, 0.15]$ be the original distribution as described by the histogram. We build a Huffman tree on P, and thus quantize the distribution to $Q = [0.125, 0.25, 0.25, 0.25, 0.125]$. The quantized distribution Q is then mapped to one of 75 Huffman trees, and can be communicated with a fixed length code of $\lceil \log_2 75 \rceil = 7$ bits.

Results. Figure 3 shows the performance of the Huffman compression scheme for the DAISY-9 configuration. The bitrate in Figure 3 is varied by increasing the number of gradient bins from 5 to 9. For the DAISY-9, VQ-7 configuration, the descriptor at 88 bits outperforms SIFT at 1024 bits.



$n = 1$ $n = 2$

$n = 3$ $n = 5$

Figure 4. Type lattices for $m = 3$

3.2. Type Coding

The idea of type coding is to construct a lattice of distributions (or *types*) $Q = Q(k_1,\ldots,k_m)$ with probabilities

$$q_i = \frac{k_i}{n}\,,\quad k_i, n \in \mathbb{Z}_+\,,\quad \sum_i k_i = n \qquad (3)$$

and then pick and transmit the index of the type that is closest to the original distribution P. Parameter n is used to control the number/density of reconstruction points. We note that type coding is related to the A_n lattice proposed in [8]. The main difference between the two is that the type lattice is naturally defined within a bounded subset of the $\mathbb{R}^m$ space, which is the unit $m-1$-simplex. This is precisely the space containing all possible input probability vectors. We show several examples of type lattices constructed for $m = 3$ and $n = 1,\ldots,5$ in Figure 4.

The total number of types in lattice (3) is essentially the number of partitions of parameter n into m terms $k_1 + \ldots + k_m = n$, given by a *multiset coefficient*:

$$\left(\!\!\binom{m}{n}\!\!\right) = \binom{n+m-1}{m-1}\,. \qquad (4)$$

Consequently, the rate needed for encoding of types satisfies:

$$R_{\text{Type}}(m,n) \leqslant \left\lceil \log_2 \left(\!\binom{m}{n}\!\right) \right\rceil \sim (m-1)\log_2 n\,. \qquad (5)$$

We next discuss quantization algorithm and combinatorial enumeration techniques needed for the design of codes.

Quantization. In order to quantize a given input distribution P to the nearest type, we use the following algorithm[1]:

1. Compute numbers (best unconstrained approximation)
$$k'_i = \left\lfloor np_i + \tfrac{1}{2} \right\rfloor\,,\quad n' = \sum_i k'_i\,.$$
2. If $n' = n$ we are done. Otherwise, compute errors
$$\delta_i = k'_i - np_i\,,$$
and sort them such that
$$-\tfrac{1}{2} \leqslant \delta_{j_1} \leqslant \delta_{j_2} \leqslant \ldots \leqslant \delta_{j_m} < \tfrac{1}{2}\,,$$
3. Let $d = n' - n$. If $d > 0$ then we decrement d values k'_i with largest errors
$$k_{j_i} = \left[\begin{array}{ll} k'_{j_i} & j = 1,\ldots,m-d-1\,, \\ k'_{j_i} - 1 & i = m-d,\ldots,m\,, \end{array}\right.$$
otherwise, if $d < 0$ we increment $|d|$ values k'_i with smallest errors
$$k_{j_i} = \left[\begin{array}{ll} k'_{j_i} + 1 & i = 1,\ldots,|d|\,, \\ k'_{j_i} & i = |d|+1,\ldots,m\,. \end{array}\right.$$

Enumeration of types. We compute a unique index $\xi(k_1,\ldots,k_m)$ for a type with coordinates $k_1,\ldots,k_m$ using:

$$\xi(k_1,\ldots,k_n) = \sum_{j=1}^{n-2} \sum_{i=0}^{k_j-1} \left(\!\!\binom{m-j}{n-i-\sum_{l=1}^{j-1} k_l}\!\!\right) + k_{n-1}\,. \qquad (6)$$

This formula follows by induction (starting with $m = 2, 3$, etc.), and it implements lexicographic enumeration of types. For example:

$$\begin{array}{rcl} \xi(0,0,\ldots,0,n) & = & 0\,, \\ \xi(0,0,\ldots,1,n-1) & = & 1\,, \\ \ldots & & \\ \xi(n,0,\ldots,0,0) & = & \left(\!\!\binom{m}{n}\!\!\right) - 1\,. \end{array}$$

This direct enumeration allows encoding/decoding operations to be performed without storing any "codebook" or "index" of reconstruction points. It can be shown that the algorithm is optimal in $||P-Q||_1$,

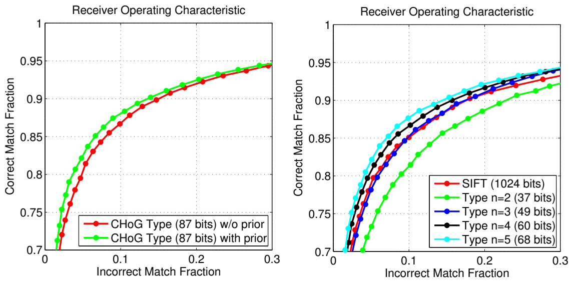
Implementation. We implement enumeration of types according to formula in Equation 6 by using an array of precomputed multiset coefficients. This reduces complexity of enumeration to just about $O(n)$ additions. In implementing type quantization, we observed that the mismatch $d = n' - n$ is typically very small, and so instead of performing full sorting step (3) we simply search for d largest or smallest numbers. With such optimization, the complexity of the algorithm becomes close to $O(m)$, instead of $O(m \log m)$ implied by the use of full search.

We also found it useful to bias type distributions as follows

$$q_i = \frac{k_i + \beta}{n + \beta m}\,. \qquad (7)$$

where parameter $\beta \geqslant 0$ is called the *prior*. The most commonly used values of β in statistics are Jeffrey's prior $\beta = 1/2$, and Laplace prior $\beta = 1$. A value of parameter β that works well is the scaled prior $\beta = \beta_0 \frac{n}{n_0}$, where n_0 is the total number of samples in the original (non-quantized)

[1] This algorithm is similar to Conway and Sloane's quantizer for A_n lattice [8], but it works within a bounded subset of $\mathbb{R}^m$



(a) (b)

Figure 5. Figure (a) shows the ROC curves of a type coded CHoG descriptor with and without priors. The performance of the descriptor is better with the scaled prior. Figure (b) shows ROC curves for compressing distributions with type coding scheme for DAISY-9 and VQ-7 configuration for *Liberty* data set. CHoG descriptor at 60 bits outperforms SIFT at 1024 bits.

histogram, and $\beta_0 = 0.5$ is the *prior* used in computation of probabilities P.

Finally, for encoding of type indices, we use both fixed-length and entropy coding schemes. We find that entropy coding with an arithmetic coder saves approximately $10-20\%$ in the bitrate. When fixed-length codes are used, we perform fast compressed domain matching.

Example. Let $m = 5$, corresponding to the VQ-5 gradient bin configuration. Let the original type described by the histogram be $T = [12, 28, 17, 27, 16]$ and $P = [0.12, 0.28, 0.17, 0.27, 0.16]$ be the corresponding distribution. Let $n = 10$ be the quantization parameter chosen for type coding. The approximation of the type T is $K = [1, 3, 2, 3, 2]$ based on Step (1) of the quantization algorithm. Since $\sum_i k_i \neq 10$, we use the proposed quantization algorithm to obtain quantized type $K = [1, 3, 2, 3, 1]$. The number of samples n_0 in the original histogram is 100, and hence, the scaled prior is computed as $\beta = 0.5 \times 10/100 = 0.05$, and the quantized distribution with prior is $Q = [0.1024, 0.2976, 0.2, 0.2976, 0.1024]$. The total number of quantized types is $\binom{14}{4} = 1001$, and Q can be communicated with a fixed length code of $\lceil \log_2 1001 \rceil = 10$ bits.

Results. Figure 5(a) illustrates the advantage of using biased types (7). Figure 5(b) shows performance of the type compression scheme for the DAISY-9, VQ-7 configuration. For this configuration, the descriptor at 60 bits outperforms SIFT at 1024 bits.

3.3. Lloyd Vector Quantization

We use Entropy Constrained Vector Quantization (ECVQ) based on the generalized Lloyd algorithm [7] to compute a bound on the performance that can be achieved with the CHoG descriptor framework. The ECVQ algorithm sweeps the optimal Rate-Distortion trade-off curve and we expect it to provide close to optimal bitrate-Equal Error Rate(EER) trade-off. The ECVQ scheme is computationally complex, and it is not practical for mobile applications. We show in Section 3.4 that the Type Coding scheme comes close to achieving the performance bound provided by ECVQ.

Quantization. The ECVQ algorithm resembles the k-means clustering in the statistics community, and, in fact, contains it as a special case. Like k-means clustering, generalized Lloyd algorithm assigns data to the nearest cluster centers, next computes new cluster centers based on this assignment, and then iterates the two steps until convergence is reached. What distinguishes the generalized Lloyd algorithm from k-means (aka the basic Lloyd algorithm) is a Lagrangian term which biases the distance measure to reflect the different number of bits required to indicate different clusters. With entropy coding, likely cluster centers will need fewer bits, while unlikely cluster centers require more bits. To properly account for bitrate, cluster probabilities are updated in each iteration of the generalized Lloyd algorithm, much like the cluster centers. We show how the ECVQ scheme can be adapted to the current CHoG framework.

Let $X^m = [p_1, p_2, p_3, ..p_m] \in \mathbb{R}^m_+$ denote a distribution. Let P_{X^m} be the distribution of X^m. Let ρ be the distance measure used to compare distributions. Let λ be the Lagrange multiplier. Let ψ be an index set, and let $\alpha :\ X^m \mapsto \psi$ quantize input vectors to indices. Let $\beta :\ \psi \mapsto C$ map indices to a set of centroids $C \in \mathbb{R}^m_+$. Let the initial size of the codebook be $K = |\psi|$. Let $\gamma(i)$ be the rate of transmitting centroid i, $i \in \psi$.

The iterative algorithm used is discussed below. The input of the algorithm is a set of points X^m, and the output is the codebook $C = \{\beta(i)\}_{i\in\psi}$. We initialize the algorithm with C as K random points and $\gamma(i) = \log_2(K)$.

1. $\alpha(x^n) = \arg\min_{i\in\psi} \rho(x^n, \beta(i)) + \lambda|\gamma(i)|$
2. $|\gamma(i)| = -\log_2 P_{X^n}(\alpha(X^n) = i)$
3. $\beta(i) = E[X^m|\alpha(X^m) = i]$

We repeat steps (1)-(3) until convergence. Step (1) is the “assignment step”, and steps (2) and (3) are the “re-estimation steps” where the centroids $\beta(i)$ and rates $\gamma(i)$ are updated. In [5], we show that comparing gradient histograms with symmetric KL divergence provides better ROC performance than using L_1 or L_2-norm. It is shown in [1, 18] that the Lloyd algorithm can be used for the general class of distance measures called Bregman divergences. Since the symmetric KL-divergence is a Bregman divergence, it can be used as the distance measure in step (1) and the centroid assignment step (3) is nevertheless optimal.

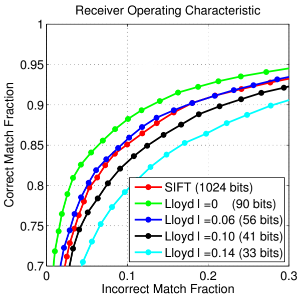


Figure 6. ROC curves for compressing distributions with Lloyd scheme for DAISY-9 and VQ-7 configuration for the *Liberty* data set. CHoG descriptor at 56 bits outperforms SIFT at 1024 bits.

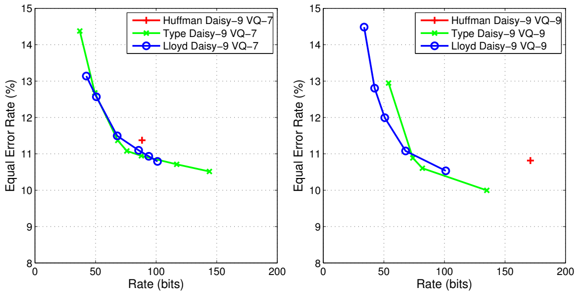
Implementation. We start with an initial codebook size of $K = 1024$ and sweep across λ to vary the bitrate for each gradient configuration. The rate decreases and the distortion increases as we increase the parameter λ. The algorithm itself reduces the size of the codebook as λ increases as certain cells become unpopulated. We add a prior of $\beta_0 = 0.5$ to all bins to avoid singularity problems. Once the histogram is quantized and mapped to an index, we entropy code the indices with an arithmetic coder. Entropy coding typically provides a $10-20\%$ reduction in bitrate compared to fixed length coding. The compression complexity of the scheme is $O(mk)$, where k is the number of cluster centroids and m is the number of gradient bins. Note that the Lloyd algorithm is computationally expensive and is not suitable for mobile applications.

Results. We show the performance of this scheme in Figure 6 for the DAISY-9, VQ-7 configuration. In Figure 6, the bitrate is varied by increasing λ with an initial codebook size of $K = 1024$. For $\lambda = 0$, we represent the descriptor with fixed-length codes in 90 bits. For this configuration, the descriptor at 56 bits outperforms SIFT at 1024 bits.

3.4. Compression Results

In this section, we compare the performance of the different histogram compression schemes. For a fair comparison at the same bit rate, we consider the Equal Error Rate (EER) point on the different ROC curves for each scheme.

First, we compare bitrate vs. EER trade-off for the different quantization schemes for DAISY-9 cell configuration in Figure 7. We observe that the Type Coding scheme outperforms the Huffman tree compresion scheme. The gain in bitrate increases as the number of gradient bins m increases. Compared to the Huffman scheme, Type Coding gives a 20% reduction in bitrate for VQ-7 at a fixed EER, and a 50% reduction in bitrate for VQ-9 at a fixed EER. Further, note that Type Coding enables a wider range of possible bitrate EER trade-offs. Next, we observe in Figures 7 and 8 that the performance of Type Coding comes close to the bound provided by Lloyd ECVQ. With Type Coding, we are able to match the performance of SIFT with about 60 bits. Finally, we compare CHoG descriptors to several other compression schemes. We consider the gradient bininng parameters VQ-3, VQ-5, VQ-7, VQ-9 and spatial binning parameters DAISY 9, 13, 17 for each quantization scheme, and compute the convex hull of the bitrate vs. EER trade-off. Figure 8 compares CHoG descriptors against the following schemes: Patch Compression with JPEG [15], Random Projections [24], Boosting Similarity Sensitive Coding [19], SIFT and SURF Transform Coding [4], Tree Structured Vector Quantization [16] and MPEG-7 Image Signatures [3]. We use the same parameters used in prior work in [5]. We observe in Figure 8 that CHoG descriptors proposed in this work outperform all other schemes.



(*a*) (*b*)

Figure 7. Figure (*a*) and (*b*) compares Huffman, Type and Lloyd Coding Schemes for DAISY-9, VQ-7 and DAISY-9 VQ-9 bin configurations respectively. The gain in bitrate for Type Coding compared to Huffman Coding increases as the number of gradient bins increases. Further, note that Type Coding comes close to achieving the bound provided by ECVQ.

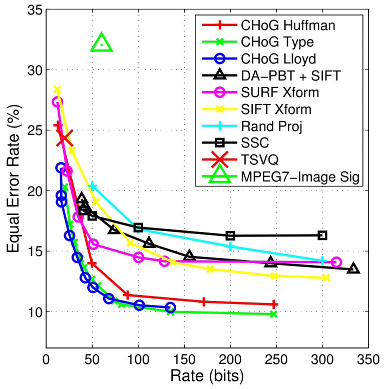


Figure 8. Comparison of EER versus bitrate for all compression schemes for the *Liberty* data set. Better performance is indicated by a lower EER. We observe that CHoG outperforms all other schemes.

4. Mobile Image Retrieval

In this Section, we show how low bit-rate CHoG descriptors enable low latency for mobile visual search applications. For such applications, one approach is to transmit the

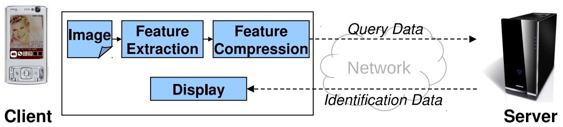


Figure 9. A mobile CD cover recognition system where the server is located at a remote location. Feature descriptors are extracted on the mobile-phone and query feature data is sent over the network.

Figure 10. A clean database picture (*top*) is matched against a real-world picture (*bottom*) with various distortions.

JPEG compressed query image over the network. An alternate approach is to extract feature descriptors on the mobile device and transmit them over the network as illustrated in Figure 9. Feature extraction can be carried out quickly (<1 second) on current generation phones making this approach feasible [20]. In this Section, we study the Classification Accuracy vs. bitrate trade-off for the two approaches.

For evaluation, we use a database of one million CD/DVD/book cover images, and a set of 1000 query images [6] exhibiting challenging photometric and geometric distortions, as shown in Figure 10. Each image has 500×500 pixels resolution. We define Classification Accuracy as the percentage of query images correctly retrieved in the top 50 images with our pipeline.

We briefly describe the retrieval pipeline for CHoG descriptors which resembles the state-of-the-art proposed in [16, 17]. We extract Difference-of-Gaussian (DoG) interest points in each image. We train a vocabulary tree [16] with depth 6 and branch factor 10, resulting in a tree with 10^6 leaf nodes. One key difference is that we use symmetric KL divergence as the distance in the clustering algorithm as KL distance performs better than L_2 norm for comparing CHoG descriptors. Since symmetric KL is a Bregman divergence [1], it can be incorporated into the k-means clustering framework. For retrieval, we use the standard TF-IDF (Term Frequency-Inverse Document Frequency) scheme [16] that represents query and database images as sparse vectors of visual word occurences, and compute a similiarity between each query and database vector. We use geometric constraints to rerank the list of top 500 images [12]. The top 50 query images are subject to pairwise matching with a RANSAC affine consistency check.

We compare three different schemes: (a) Transmitting JPEG compressed images, (b) Transmitting SIFT descriptors and (c) Transmitting CHoG descriptors. Figure 11 shows the performance of the three schemes. For Scheme (a), we transmit a 480×480 gray-scale JPEG compressed image accross the network. The bitrate is varied by changing the quality of JPEG compression. Feature extraction and matching are carried out on the JPEG compressed image on the server. We observe that the performance of the scheme deteriorates rapidly at low bitrates. At low bitrates, interest point detection fails due to blocking artifacts introduced by JPEG image compression.

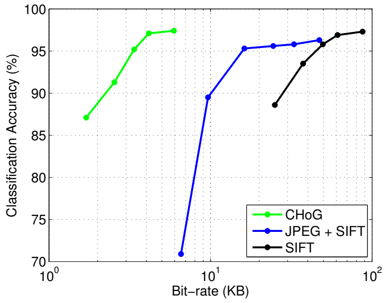


Figure 11. Retrieval results for a database containing 1 million images. We achieve 96 percent accuracy with only 4 kilobytes of data. Note that the retrieval performance of CHoG is similar to SIFT and JPEG compression schemes, while the bitrate savings is 16× and 10× respectively.

Scheme	Upload Time (20 kbps)	Upload Time (60 kbps)
JPEG+SIFT	20.0	6.7
SIFT	32.0	10.7
CHoG	1.6	0.5

Table 1. Transmission times for different schemes at varying network uplink speeds

For Schemes (b) and (c), we extract descriptors on the mobile device and transmit them over the network. The bitrate is varied by varying the number of descriptors from 200 to 700. We pick the features with the highest Hessian response [14] for a given feature budget. We observe that transmitting 1024-bit SIFT descriptors is almost always more expensive than transmitting the entire JPEG compressed image. For Scheme (c), we use a low bit-rate Type coded CHoG descriptor. We use spatial bin configuration DAISY-9, gradient bin configuration VQ-7 and type coding parameter $n = 7$, which generates a ~70 bit descriptor. We observe that CHoG descriptor achieves a comparable Classification Accuracy to SIFT with bitrate savings of 16×. Compared to sending JPEG compressed images, CHoG descriptor achieves bitrate savings of 10×. We compare transmission times for typical cellular uplink speeds in Table 1 for the different schemes. At 20 kbps, the difference in latency between CHoG and the other schemes is about 20 seconds. We conclude that transmitting CHoG descriptors reduces query latency significantly for mobile visual search applications.

5. Conclusion

We study different quantization schemes for the Compressed Histogram of Gradients (CHoG) image feature descriptor. We propose a scheme for compressing distributions called Type Coding, which offers lower complexity and higher compression efficiency compared to tree-based quantization schemes. We construct optimal Entropy Constrained Vector Quantization (ECVQ) code-books and show that Type Coding comes close to achieving optimal performance. The proposed descriptors are 16× smaller than SIFT and can be compared in the compressed domain. We implement the descriptor in a mobile image retrieval system, and for a database of 1 million CD, DVD and book covers, we achieve 96% retrieval accuracy with only 4 kilobytes of data per query image.

References

[1] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman divergences. In *Journal of Machine Learning Research*, pages 234–245, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[3] P. Brasnett and M.Z.Bober. Robust visual identifier using the trace transform. In *Proc. of IET Visual Information Engineering Conference (VIE)*, London, UK, July 2007.

[4] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, and B. Girod. Transform coding of feature descriptors. In *Proc. of Visual Communications and Image Processing Conference (VCIP)*, San Jose, California, January 2009.

[5] 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 *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, Florida, June 2009.

[6] D. M. Chen, S. S. Tsai, R. Vedantham, R. Grzeszczuk, and B. Girod. *CD Cover Database - Query Images*, April 2008.

[7] P. A. Chou, T. Lookabaugh, and R.M.Gray. Entropy constrained vector quantization. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(1), Jan 1989.

[8] J. H. Conway and N. J. A. Sloane. Fast quantizing and decoding algorithms for lattice quantizers and codes. IT-28(2):227–232, Mar. 1982.

[9] T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[10] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005.

[11] G. Hua, M. Brown, and S. Winder. Discriminant Embedding for Local Image Descriptors. In *Proc. of International Conference on Computer Vision (ICCV)*, 2007.

[12] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 304–317, Berlin, Heidelberg, 2008.

[13] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 02, pages 506–513. IEEE Computer Society, 2004.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[15] M. Makar, C. Chang, D. M. Chen, S. S. Tsai, and B. Girod. Compression of Image Patches for Local Feature Extraction. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009.

[16] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, USA, June 2006.

[17] 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. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, June 2008.

[18] D. Rebollo-Monedero. Quantization and Transforms for Distributed Source Coding. *Ph.D. thesis, Deparment of Electrical Engineering, Stanford University*, 2007.

[19] G. Shakhnarovich and T. Darrell. Learning Task-Specific Similarity. *Thesis*, 2005.

[20] G. Takacs, V. Chandrasekhar, D. M. Chen, S. S. Tsai, R. Grzeszczuk, and B. Girod. Unified Real-time Tracking and Recognition with Rotation Invariant Fast Features. In *Accepted to IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, SFO, California, June 2010.

[21] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod. Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In *Proc. of ACM International Conference on Multimedia Information Retrieval (ACM MIR)*, Vancouver, Canada, October 2008.

[22] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[23] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, Miami, Florida, June 2009.

[24] C. Yeo, P. Ahammad, and K. Ramchandran. Rate-efficient visual correspondences using random projections. In *Proc. of IEEE International Conference on Image Processing (ICIP)*, San Diego, California, October 2008.