

Adapting Information Theoretic Clustering to Binary Images

Christian Bauckhage^{1,2}
¹*B-IT, University of Bonn*
 53113 Bonn, Germany
<http://www.b-it-center.de>

Christian Thureau²
²*Fraunhofer IAIS*
 53754 St. Augustin, Germany
<http://www.iais.fraunhofer.de>

Abstract

We consider the problem of finding points of interest along local curves of binary images. Information theoretic vector quantization is a clustering algorithm that shifts cluster centers towards the modes of principal curves of a data set. Its runtime characteristics, however, do not allow for efficient processing of many data points. In this paper, we show how to solve this problem when dealing with data on a 2D lattice. Borrowing concepts from signal processing, we adapt information theoretic clustering to the quantization of binary images and gain significant speedup.

1 Introduction

In a project on analyzing large amounts of digitized graffiti tags as shown in Figure 1, we were faced with the problem of locating interest points along the strokes of highly stylized letters. We tested morphological operators, medial axis transforms, key point detectors, and clustering algorithms. Information theoretic clustering was found to yield good results, but in its original form, it proved too slow to be of practical use.

In this paper, we present a novel variant of information theoretic vector quantization that is tailored to the quantization of binary images. Recasting essential steps of the original method in terms of efficient convolution operations significantly reduces runtime and enables us to quickly extract interest points from large amounts of shape images.

Next, we briefly review information theoretic clustering and discuss its properties. Then we derive our accelerated version and present experimental evaluations that underline its favorable performance.

2 Information Theoretic Clustering

Given a set of data $\mathcal{X} = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$, information theoretic vector quantization (ITVQ) as intro-

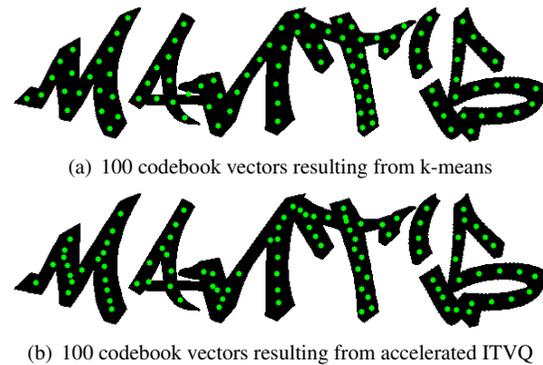


Figure 1. Different placements of codebook vectors on a digitized graffiti tag. While k-means evenly distributes the cluster centers, ITVQ shifts them towards local principal curves.

duced in [4, 5, 6] considers an entropy-based measure to determine a suitable set of $M \ll N$ codebook vectors $\mathcal{W} = \{w_1, \dots, w_M\} \subset \mathbb{R}^d$.

Assuming that probability density functions $p(x)$ and $q(x)$ are available that characterize \mathcal{X} and \mathcal{W} , respectively, ITVQ iteratively optimizes the location of the codebook vectors by means of minimizing the Cauchy-Schwartz divergence

$$D_{cs}(\mathcal{X}, \mathcal{W}) = 2H(\mathcal{X}; \mathcal{W}) - H(\mathcal{W}) - H(\mathcal{X})$$

between $p(x)$ and $q(x)$. Note that the last term of D_{cs} does not depend on \mathcal{W} ; the entropies in the first and second term are defined as

$$\begin{aligned} H(\mathcal{X}; \mathcal{W}) &= -\log \int p(x)q(x)dx \\ &= -\log V(\mathcal{X}; \mathcal{W}) \\ H(\mathcal{W}) &= -\log \int q^2(x)dx \\ &= -\log V(\mathcal{W}) \end{aligned}$$

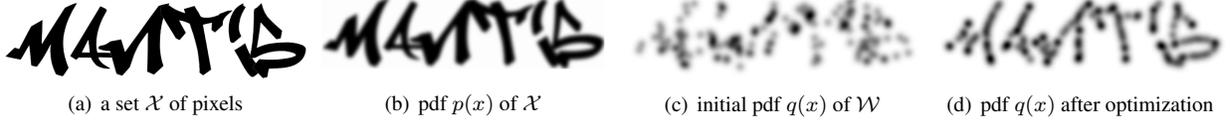


Figure 2. Information theoretic clustering for shape quantization: (a) a binary shape forms a set \mathcal{X} of 2D vectors x_i ; (b) the Parzen estimate of the pdf $p(x)$ of \mathcal{X} ; (c) for randomly initialized codebook vectors w_j , the pdf $q(x)$ of \mathcal{W} diverges from $p(x)$; (d) minimizing the Cauchy-Schwartz divergence of $p(x)$ and $q(x)$ shifts the codebook vectors towards (local) principal curves of \mathcal{X} .

and correspond to Renyi's cross entropy between $p(x)$ and $q(x)$ and Renyi's entropy of $q(x)$, respectively. The fundamental idea in [4, 5, 6] is to model the densities $p(x)$ and $q(x)$ by means of their Parzen estimates

$$\begin{aligned} p(x) &= \frac{1}{N} \sum_{i=1}^N G_{\xi}(x - x_i) \\ q(x) &= \frac{1}{M} \sum_{j=1}^M G_{\omega}(x - w_j) \end{aligned} \quad (1)$$

where we use the shorthand $G_{\sigma}(x) = \exp -\frac{\|x\|^2}{2\sigma^2}$ to denote Gaussian kernels of variance σ^2 .

Because of the Gaussian product theorem and properties of the so called overlap integral, the entropies can then be written as

$$\begin{aligned} \int p(x)q(x)dx &= \frac{\sum_{i,j} \int G_{\xi}(x - x_i)G_{\omega}(x - w_j)dx}{MN} \\ &= \frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M G_{\tau}(x_i - w_j) \\ \int q^2(x)dx &= \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M G_{\rho}(w_i - w_j) \end{aligned}$$

where $\tau^2 = \xi^2 + \omega^2$ and $\rho^2 = 2\omega^2$.

Differentiating $D_{cs}(\mathcal{X}, \mathcal{W})$ with respect to w_k , equating to zero, and rearranging the resulting terms yields a fix point update rule for the vectors in \mathcal{W}

$$\begin{aligned} w_k^{\text{new}} &= \frac{\sum_{j=1}^N G_{\tau}(x_j - w_k)x_j}{\sum_{j=1}^N G_{\tau}(x_j - w_k)} \\ &- c \cdot \frac{\sum_{j=1}^M G_{\rho}(w_j - w_k)w_j}{\sum_{j=1}^N G_{\tau}(x_j - w_k)} \\ &+ c \cdot \frac{\sum_{j=1}^M G_{\rho}(w_j - w_k)}{\sum_{j=1}^N G_{\tau}(x_j - w_k)} w_k \end{aligned} \quad (2)$$

where the constant c is given by $\frac{N}{M} \frac{V(\mathcal{X}; \mathcal{W})}{V(\mathcal{W})}$.

It can be shown that the mean shift procedure [1, 2] is a special case of ITVQ [5]. The cross entropy term $\log V(\mathcal{X}; \mathcal{W})$ in the Cauchy-Schwartz divergence turns ITVQ into a mode seeking algorithm. The quantity $\log V(\mathcal{W})$, on the other hand, can be understood as the potential of a repellent force between codebook vectors. The resulting codebook vectors are thus located at local modes, but in contrast to mean shift, the repellent force prevents them from collapsing into a few modes only. Figure 2 illustrates the process of ITVQ and how traces out local principal curves of binary shapes.

For N data points and M codebook vectors, the computing time per iteration is of the order $O(MN)$. However, in contrast to other clustering algorithms of the same complexity, such as k-means, there is a large constant that factors into the overall runtime: each of the $k = 1, \dots, M$ updates in (2) requires the computation of $N+M$ distances $\|x_j - w_k\|^2$ or $\|w_j - w_k\|^2$ as well as the evaluation of as many $\exp(\cdot)$ functions. In practice, we found the cost of a single iteration to be prohibitive even for moderately sized shape images.

3 Accelerated ITVQ

In this section, we derive an update rule for information theoretic vector quantization that is tailored to binary images. The key idea is to model Parzen density estimates using methods from signal processing. This eliminates the need of having to compute many distances and exponentials and thus considerably accelerates the procedure.

First, we note that convolving a function f with a shifted delta impulse shifts the function

$$\delta(x - \mu) * f(x) = f(x - \mu).$$

A Parzen density estimate using a sum of Gaussians centered at data points x_j thus equals a sum of shifted delta impulses convolved with a Gaussian

$$\sum_j G_{\sigma}(x - x_j) = G_{\sigma}(x) * \sum_j \delta(x - x_j).$$

Key to our algorithm is that every pixel of a binary image can be seen as a discrete delta impulse located at some point on a regular 2D lattice. In a slight abuse of notation, we may therefore write the density function of a set \mathcal{X} of pixels as

$$p(x) = \mathcal{X} * G_\xi.$$

Convolution with a Gaussian is of course a standard operation in image processing. If ξ is small, the 2D convolution can be efficiently separated into two 1D convolutions with corresponding, pre-computed filter masks; if ξ is large, it is more efficient to apply recursive schemes [3, 7]. Although this way of computing $p(x)$ will still require efforts proportional to the number N of pixels, the effort per pixel will become very small.

If the codebook \mathcal{W} , too, is understood as a binary image, i.e. as an image where there are M active pixels, its density $q(x) = \mathcal{W} * G_\omega$ can be computed just as efficiently. On a discrete lattice such as a pixel array, the entropy integrals $V(\mathcal{X}; \mathcal{W}) = \int p(x)q(x)dx$ and $V(\mathcal{W}) = \int q^2(x)dx$ are therefore readily available.

It remains to derive an update rule similar to (2). To this end, we once again consider (1) and note that

$$\begin{aligned} \frac{\partial q(x)}{\partial w_k} &= \frac{\partial}{\partial w_k} \frac{1}{M} \sum_{j=1}^M G_\omega(x - w_j) \\ &= \frac{1}{M} \frac{w_k - x}{2\omega^2} G_\omega(x - w_k). \end{aligned}$$

Using this, we differentiate $D_{cs}(\mathcal{X}, \mathcal{W})$ with respect to w_k , equate to zero, rearrange the resulting terms, and obtain the following update rule:

$$\begin{aligned} w_k^{\text{new}} &= \frac{\int p(x)G_\omega(x - w_k)xdx}{\int p(x)G_\omega(x - w_k)dx} \\ &- \frac{V(\mathcal{X}; \mathcal{W})}{V(\mathcal{W})} \frac{\int q(x)G_\omega(x - w_k)xdx}{\int p(x)G_\omega(x - w_k)dx} \\ &+ \frac{V(\mathcal{X}; \mathcal{W})}{V(\mathcal{W})} \frac{\int q(x)G_\omega(x - w_k)dx}{\int p(x)G_\omega(x - w_k)dx} w_k. \quad (3) \end{aligned}$$

In contrast to (2), every Gaussian in (3) is of the form $\delta(x - w_k) * G_\omega(x)$. This allows for further acceleration. Following common image processing practice, we approximate the Gaussian $G_\omega(x)$ by means of a pre-computed, discrete filter mask F of finite support. Instead of evaluating the integrals in (3) over the whole image plane, it then suffices to consider a neighborhood \mathcal{N} of w_k whose size depends on ω ; for instance:

$$\int p(x)G_\omega(x - w_k)xdx \approx \int_{\mathcal{N}} p(x)F(x - w_k)xdx$$

Concluding our derivation, we observe that, in contrast to the original update algorithm for ITVQ, our ver-

Input: discrete 2D vectors $\mathcal{X} \Leftrightarrow$ binary image \mathbf{X}
Output: discrete codebook $\mathcal{W} \Leftrightarrow$ binary image \mathbf{W}

initialize \mathbf{W} through sampling the active pixels of \mathbf{X}

approximate the pdf $p(x)$ using $\mathbf{P} = \mathbf{X} * G_\xi$

$t = 0$

repeat

approximate the pdf $q(x)$ using $\mathbf{Q} = \mathbf{W} * G_\omega$

compute $V(\mathcal{X}; \mathcal{W}) = \sum_{i,j} P_{ij}Q_{ij}$

compute $V(\mathcal{W}) = \sum_{ij} Q_{ij}^2$

for $k = 1, \dots, M$

compute w_k^{new} using (3)

update $\mathcal{W} = \bigcup_k w_k^{\text{new}}$

until $D_{cs}(\mathcal{X}; \mathcal{W}) \leq \epsilon \vee t \leftarrow t + 1 > t_{\max}$

Figure 3. Accelerated ITVQ tailored to the processing of binary images.

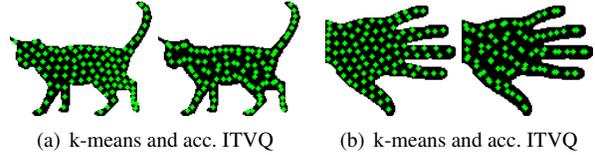


Figure 4. Locations of 100 codebook vectors on shapes from the Kimia data set.

sion for binary images avoids (i) computation of Euclidean distances and (ii) explicit evaluation of exponentials. Rather, it solely resorts to convolutions which can either be efficiently computed for the whole image or may be evaluated from finite local support only. Figure 3 summarizes the accelerated ITVQ algorithm.

4 Practical Performance

In this section, we present and discuss results obtained from experimenting with the original ITVQ algorithm and our accelerated version. We also discuss results from using k-means clustering for binary image quantization to provide a baseline for comparison.

All experiments were carried out on an Intel Core 2 Duo CPU (2.53 GHz). The variance parameter ω was set dynamically, taking into account the number N of pixels in a shape and the number M of codebook vectors to be produced: $\omega = \sqrt{N/M}/2$. This way, a smaller codebook will lead to larger discrete fil-

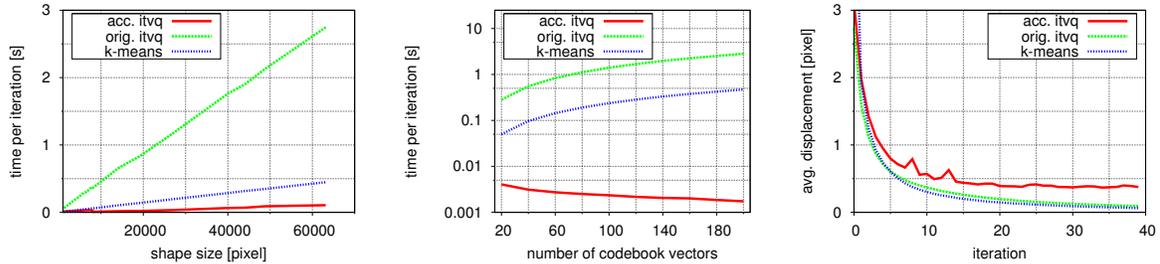


Figure 5. Left panel: average computing time per iteration w.r.t. shape size. The graphs compare the performance of accelerated ITVQ, original ITVQ, and k-means in extracting 100 codebook vectors from shapes of growing size. Middle panel: average computing time per iteration w.r.t. number of codebook vectors extracted from a shape of about 8800 pixels. Original ITVQ and k-means show linear growths (note the logarithmic scale); for accelerated ITVQ, our choice of local support regions (see text) even causes decreasing runtime. Right panel: average displacement of a codebook vector w.r.t. number of iterations. Original ITVQ and k-means perform continuous optimization and continue to slightly shift codebook vectors for many iterations. Accelerated ITVQ optimizes the location of codebook vectors on a discrete lattice; after about 15 iterations, codebook vectors jitter by less than half a pixel but cease to shift.

ter masks applied in computing (3). Note that in order to benefit from our idea of local support, it may in fact be preferable to consider convolution masks of smaller diameter. The parameter ξ was set to $\omega/2$. Finally, all presented results are averaged over multiple trials with random initializations of \mathcal{W} .

Figures 1 and 4 exemplify the different placements of codebook vectors produced by k-means and accelerated ITVQ. Where k-means produces almost uniformly distributed coverings of the underlying shapes, ITVQ places the codebook vectors at local modes of the (shape) images. This causes the codebook vectors to better trace out elongated parts of a shape.

The left and middle panel of Figures 5 underline the runtime improvement of accelerated ITVQ over the original algorithm. For larger choices of N and M , the time per iteration required by our accelerated version of ITVQ is an order of magnitude smaller than that required by the original version. The declining time per iteration of accelerated ITVQ for a growing number of clusters is a consequence of the above choice of ω and its impact on the size of filter masks.

The right panel in Fig. 5 shows the convergence behavior of the three methods. In contrast to k-means and the original ITVQ procedure, the accelerated algorithm is tailored to data on a discrete lattice and uses convolution masks of finite support. The resulting round offs cause the average shift of codebook vectors not to decrease to zero but to jitter. In this sense the algorithm converges, once the average displacement of codebook vectors falls below 0.5 pixels. Applications that do not require subpixel accuracy but real time capability do

greatly benefit from this characteristic: convergence requires only a few iterations and since each iteration only requires fractions of a second, the proposed version of ITVQ allows for processing several shapes per second.

5 Summary

We introduced an accelerated version of information theoretic vector quantization that efficiently computes interest points along the modes of principal curves of shape images. Replacing costly computations in the original algorithm through efficient convolution operations, we achieved significant speedup.

References

- [1] Y. Cheng. Mean Shift, Mode Seeking, and Clustering. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(8), 1995.
- [2] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach towards Feature Space Analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(5), 2002.
- [3] R. Deriche. Recursively Implementing the Gaussian and its Derivatives. In *Proc. ICIP*, 1992.
- [4] T. Lehn-Schioler, A. Hedge, D. Erdogmus, and J. Principe. Vector Quantization Using Information Theoretic Concepts. *Natural Computing*, 4(1), 2005.
- [5] S. Rao, A. de Medeiros Martins, and J. Principe. Mean Shift: An Information Theoretic Perspective. *Pattern Recognition Letters*, 30(3), 2009.
- [6] S. Rao, S. Han, and J. Principe. Information Theoretic Vector Quantization with Fixed Point Updates. In *Proc. IJCNN*, 2007.
- [7] L. van Vliet, I. Young, and P. Verbeek. Recursive Gaussian Derivative Filters. In *Proc. ICPR*, 1998.