Mapping Natural Image Patches by Explicit and Implicit Manifolds

Kent Shi and Song-Chun Zhu
UCLA Department of Statistics
{kentshi, sczhu}@stat.ucla.edu

Abstract

Image patches are fundamental elements for object modeling and recognition. However, there has not been a panoramic study of the structures of the whole ensemble of natural image patches in the literature. In this article, we study the structures of this ensemble by mapping natural image patches into two types of subspaces which we call "explicit manifolds" and "implicit manifolds" respectively. On explicit manifolds, one finds those simple and regular image primitives, such as edges, bars, corners and junctions. On implicit manifolds, one finds those complex and stochastic image patches, such as textures and clutters. On different types of manifolds, different perceptual metrics are used. We propose a method for learning a probabilistic distribution on the space of patches by pursuing both types of manifolds using a common information theoretical criterion. The connection between the two types of manifolds is realized by image scaling, which changes the entropy of the image patches. The explicit manifolds live in low entropy regimes while the implicit manifolds live in high entropy regimes. We study the transition between the two types of manifolds over scale and show that the complexity of the manifolds peaks in a middle entropy regime.

1. Introduction

Image patches at multiple resolutions are fundamental elements for object recognition. Recently, a number of patch-based methods have been proposed in the literature [5, 6, 8, 13]. Meanwhile, different theories have been developed for modeling natural image patches, including sparse coding models [12] and Markov random fields [15]. However, there has not been a panoramic study of the structures of the whole ensemble of natural image patches, except some recent attempts to calculate the statistics of 3×3 patches in natural images [7, 3]. Such a panoramic point of view is useful because it enables us to view different models simply as different manifolds in the space of image patches, so that these models and concepts can be pursued in a common framework.

To be more specific, we argue that the two classes of models – the generative sparse coding models and the descriptive Markov random fields [15] are two different ways of representing and mapping natural image patches with different metrics for different purposes.

Sparse coding models for geometric primitives [12]: These models represent image patches by an image generating function parameterized by a small number of *hidden variables* indexing the photometric and geometric properties of the image patches. By varying the values of these variables, the model generates a set of image patches that span a low-dimensional manifold in the space of image patches. We call this manifold the *explicit* manifold, because the image patches on this manifold can be accurately mapped and reconstructed explicitly by the corresponding values of the variables in the model. On explicit manifolds, we usually find simple and regular image patches such as edges, bars, corners, junctions, and other geometric primitives. The left picture of Fig. 2 illustrates the explicit manifolds, where each image patch is a point. An explicit manifold can be a zero-dimensional point, one-dimensional line, or a two-dimensional surface etc. Two image patches are considered similar if their values of hidden variables are close to each other.

Markov random fields for stochastic textures [15]: These models represent image patches by a small number of *feature statistics* indexing the texture properties of the image patches. Two image patches have similar texture properties as long as the values of their feature statistics are close to each other, even though they may differ greatly in image intensities. The set of image patches that share the same value of feature statistics form an *implicit* manifold, because these image patches cannot be explicitly reconstructed by the feature statistics, which only impose some constraints. On implicit manifolds, we usually find complex and stochastic image patches such as textures and clutters.

In the space of image patches, implicit manifolds have higher dimensions and often submerge explicit manifolds. By analogy to cosmology, the distribution of natural image patches is similar to the distribution of mass in the universe as shown in Fig. 1. The image patch space has many low

1-4244-1180-7/07/$25.00 ©2007 IEEE

Figure 1. The distribution of natural image patches is similar to the distribution of mass in universe, where there are high density and low volume stars as well as low density and high volume nebulae.

dimensional explicit manifolds with high densities, they are like the bright stars in Fig. 1. For example, a step edge is a low dimensional manifold in the image space [7]. There are also many high dimensional implicit manifolds, they are like the nebulae in Fig. 1. For example, the texture patches of sky, wall, floor, foliage etc. live on high dimensional implicit manifolds.

The mixing of these manifolds make the clustering and learning tasks difficult. Recently, there has been some work on clustering data by generalized PCA [9], which assumes linear subspaces, which are explicit manifolds. In the literature of vision and machine learning, there has been no previous work that learns the explicit and implicit manifolds simultaneously. Although some models like Mumford-Shah [11] incorporate both Markov random fields and edge primitives, there is neither theoretical justification for such models nor information theoretical principles for learning them.

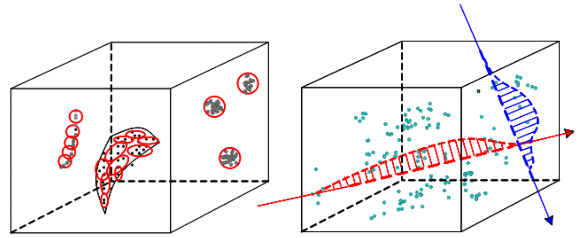
From the panoramic point of view, different image models correspond to different manifolds in the same space of image patches. Thus, we can pursue different models in a common framework of selecting manifolds to model the ensemble of natural image patches by minimizing the Kullback-Leibler divergence. This gives a theoretical justification why we need hidden variables and feature statistics to characterize natural image patches.

We also study the connection between the two types of manifolds in terms of scale or resolution. We show that image patches corresponding to different scales or resolutions should be fitted by different types of manifolds, and the complexity of the manifolds peaks at medium resolution, which is considered the most informative resolution for object recognition.

2. Two types of manifolds and image modeling

2.1. Explicit and implicit manifolds

Consider image patches $\mathbf{I}$ defined on a domain D (e.g., 20×20 lattice) with $|D|$ pixels. Let $\Omega = [1, L]^D$ be the set of all image patches, where the grey levels of $\mathbf{I}$ take integer values from 1 to L. Ω is the space of image patches.



(a) explicit manifolds (b) implicit manifold

Figure 2. Illustration of explicit manifolds and implicit manifolds in the space of image patches, where each image patch is a point. In the left figure, an explicit manifold can be a low-dimensional surface. In the right figure, the image patches are mapped to feature statistics such as marginal histograms that constrain implicit manifolds.

Definition: *An explicit manifold is defined as*

$$\Omega^{\text{ex}} = \{\mathbf{I} : \mathbf{I} = \Phi(w), \forall w \in W\}, \tag{1}$$

where $\Phi(w)$ is an explicit image generating function, and $\mathbf{I} = \Phi(w)$ means both sides are equal up to discretization accuracy. w is a low-dimensional hidden variable taking values in a set W. w usually includes both photometric and geometric properties of the image patches. See Fig. 2.(a) for an illustration of the explicit manifolds.

An example of $\Phi(w)$ is edge and bar model [2]. An edge patch is modeled by a function whose profile is a step function blurred by a Gaussian kernel. The photometric components of w include the intensities on the two sides of the step function, as well as the standard deviation of the Gaussian kernel. The geometric components of w include location, orientation, and length of the function. A bar patch is modeled by a function whose profile has three constant fragments. The edge and bar model can be further composed into corners, junctions and crosses etc [4].

There can be a large number of primitives, which correspond to a large number of explicit manifolds $\Omega^{\text{ex}}_m, m = 1, ..., M$.

Definition: *An implicit manifold is defined as*

$$\Omega^{\text{im}} = \{\mathbf{I} : \mathbf{h}(\mathbf{I}) = \mathbf{h}\}, \tag{2}$$

where $\mathbf{h}(\mathbf{I}) = \frac{1}{|D|}\sum_{x \in D} F_x(\mathbf{I})$ is the feature statistics pooled over the patch for some feature extractor F. Usually $\mathbf{h}(\mathbf{I})$ is the marginal histograms of filter responses or local orientations, see Fig. 2.(b) for an illustration.

An implicit manifold is indexed by $\mathbf{h}$. Asymptotically, the uniform distribution over Ω^{im} defined by (2) is equivalent to the following Markov random field model [15]

$$p(\mathbf{I}; \mathbf{h}) = \frac{1}{Z(\lambda)} \exp\{\langle \lambda, \mathbf{H}(\mathbf{I}) \rangle\}, \tag{3}$$

where $Z(\lambda)$ is the normalizing constant, and $\mathbf{H}(\mathbf{I}) = \sum_x F_x(\mathbf{I})$. λ is calculated so that $\mathrm{E}_\lambda[\mathbf{h}(\mathbf{I})] = \mathbf{h}$. The reason for this asymptotical equivalence is that as $|D| \to \infty$, $\mathbf{H}(\mathbf{I})/|D|$ converges to a constant due to ergodicity, and p($\mathbf{I}$; $\mathbf{h}$) is constant for all those $\mathbf{I}$ with the same $\mathbf{H}(\mathbf{I})$.

There can be a large number of Markov random fields or feature statistics, which correspond to a large number of implicit manifolds $\Omega_m^{\text{im}}, m = 1, ..., M$.

2.2. Image modeling and KL divergence

Let $f(\mathbf{I})$ be the frequency distribution of the whole ensemble of image patches over Ω. The goal of visual learning is to learn a statistical model $p(\mathbf{I})$ to approximate $f(\mathbf{I})$, by minimizing the Kullback-Leibler divergence

$$\mathcal{D}(f||p) = \mathrm{E}_f[\log \frac{f(\mathbf{I})}{p(\mathbf{I})}] = -\mathrm{E}_f[\log p(\mathbf{I})] + \text{const} \tag{4}$$

within a class $\mathcal{M}$ of candidate distributions or models for p. In Eqn. (4), $\mathrm{E}_f[\log p(\mathbf{I})]$ is the population-wise log-likelihood of p. In practice, if we observe a training sample $\mathbf{I}_j \sim f, j = 1, ..., n$, we can approximate

$$\mathrm{E}_f[\log p(\mathbf{I})] \approx \frac{1}{n} \sum_{j=1}^{n} \log p(\mathbf{I}_j). \tag{5}$$

So minimizing Kullback-Leibler divergence is asymptotically equivalent to maximizing log-likelihood. The Kullback-Leibler divergence also measures the redundancy of coding f based on p.

The learning can be a sequential process, which pursues the model in a sequence of model spaces $\mathcal{M}_0 \subset \mathcal{M}_1 \subset ... \subset \mathcal{M}_K \subset ...$ of increasing complexities. At each step, we augment the model by introducing new structures or features to minimize the Kullback-Leibler divergence.

3. Manifold pursuit

There can be a large number of candidate sparse coding models or Markov random fields, which correspond to different explicit and implicit manifolds. In order to pursue these manifolds in a common framework, we need to build a model $p(\mathbf{I})$ based on these manifolds, so that in the context of this model, we can sequentially single out the manifolds by minimizing the Kullback-Leibler divergence $\mathcal{D}(f||p)$ in order to efficiently code f. The selected manifolds then give rise to different models for image patches.

Specially, let $\Omega_k, k = 1, ..., K$ be the K manifolds to be chosen from a large collection of candidate explicit or implicit manifolds. Let $\Omega_0 = \Omega \backslash \bigcup_{k=1}^K \Omega_k$, we use the following model to choose $\{\Omega_k\}$:

$$p(\mathbf{I}; \gamma) = \frac{1}{Z(\gamma)} \exp\{\sum_{k=0}^{K} \gamma_k 1_{\Omega_k}(\mathbf{I})\}, \tag{6}$$

where $1_{\Omega_k}(\mathbf{I})$ is the indicator function, which equals 1 if $\mathbf{I} \in \Omega_k$, and 0 otherwise. $Z(\gamma)$ is the normalizing constant. This model can be considered a panoramic approximation to $f(\mathbf{I})$ based on $\{\Omega_k\}$. It is a special case of [1].

Model (6) seeks to match the frequencies of the manifolds in the ensemble of natural image patches f. Specifically, let

$$f_k = \mathrm{E}_f[1_{\Omega_k}(\mathbf{I})] = \Pr(\mathbf{I} \in \Omega_k). \tag{7}$$

f_k can be estimated from the training examples by the corresponding frequencies. If $\hat{\gamma}$ minimizes $\mathcal{D}(f||p)$ over all possible values of γ, then it can be shown that $\mathrm{E}_{\hat{\gamma}}[1_{\Omega_k}(\mathbf{I})] = f_k$. Model (6) is the maximum entropy model in that among all the probability distributions p such that $\mathrm{E}_p[1_{\Omega_k}(\mathbf{I})] = f_k$, $p(\mathbf{I}; \hat{\gamma})$ has the maximum entropy. This means that after matching the frequencies f_k, we leave the probability distribution to be as smooth as possible within Ω_k or their interactions.

Recall that each Ω_k corresponds to a sparse coding model or a Markov random field, so model (6) can be considered a meta-model, or a model of models, because it is built on $\{\Omega_k\}$. The pursuit of different types of Ω_k reveal the origins of different types of models.

In the context of model (6), we may pursue $\Omega_k, k = 1, ..., K$ by sequentially minimizing the corresponding $\mathcal{D}(f||p(\mathbf{I}; \gamma))$, i.e., at each step, we choose Ω_k that leads to the maximum reduction of $\mathcal{D}(f||p(\mathbf{I}; \gamma))$. Specifically, let $p(\mathbf{I}; \hat{\gamma})$ be the currently fitted model, and we want to introduce a new manifold Ω_{K+1} to augment the model to a new fitted model $p(\mathbf{I}; \hat{\gamma}_+)$, with $\gamma_+ = (\gamma, \gamma_{K+1})$. Then we can define the information gain of Ω_{K+1} as

$$\mathcal{D}(f||p_{\hat{\gamma}}) - \mathcal{D}(f||p_{\hat{\gamma}_+}) = \mathcal{D}(p_{\hat{\gamma}_+}||p_{\hat{\gamma}}). \tag{8}$$

If Ω_{K+1} is an explicit manifold, (8) measures the information gain by adding a hidden variable or a new structure. If Ω_{k+1} is an implicit manifold, (8) measures the information gain by adding a feature statistics or a new set of feature statistics.

If Ω_k are non-overlapping, model (6) reduces to

$$p(\mathbf{I}) = \sum_{k=0}^{K} f_k U[\Omega_k], \tag{9}$$

where $U[\Omega_k]$ is the uniform distribution over Ω_k. This model is often a reasonable approximation to model (6).

For model (9), the Kullback-Leibler divergence is

$$\mathcal{D}(f||p) = -\sum_{k=0}^{K} f_k \log \frac{f_k}{|\Omega_k|} + \mathrm{E}_f[\log f(\mathbf{I})], \tag{10}$$

so we can measure the information gain of Ω_k by

$$l_k = f_k[\log f_k - \log(|\Omega_k|/|\Omega|)], \tag{11}$$

and pursuing Ω_k according to l_k.

Pursuit of implicit manifolds: If $\Omega_k = \{\mathbf{I} : \mathbf{H}_k(\mathbf{I})/|D| = \mathbf{h}_k\}$. Under the uniform distribution over Ω, $\mathbf{H}_k(\mathbf{I})/|D|^{1/2}$ converges to a multivariate Gaussian distribution $\mathrm{N}(\mathbf{h}_0, \Sigma_0)$ according to the central limit theorem, so approximately

$$\log |\Omega_k|/|D| \approx \log L - (\mathbf{h}_k - \mathbf{h}_0)'\Sigma_0^{-1}(\mathbf{h}_k - \mathbf{h}_0)/2, \quad (12)$$

where L is the number of grey levels. (12) is computable and can be used with (10) and (11) to add new feature statistics sequentially.

Pursuit of explicit manifolds: If $\Omega_k = \{\mathbf{I} : \mathbf{I} = \Phi_k(w_k)\}$, with $w_k = (w_{k,1}, ..., w_{k,d})$, then

$$\log |\Omega_k| = \sum_{i=1}^{d} \log L_i, \quad (13)$$

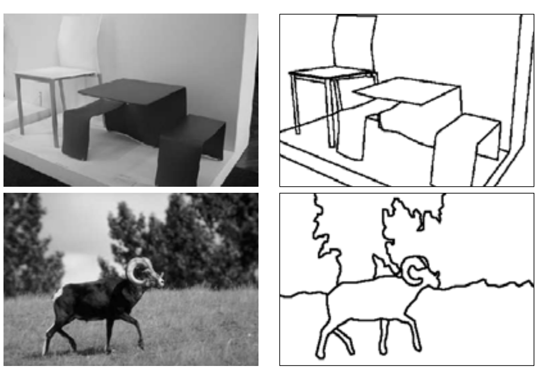
where L_i is the number of discretization levels of $w_{k,i}$. (13) can be used with (10) and (11) to add new variables sequentially.

The explicit and implicit manifolds work along opposite directions in the following sense. By augmenting new hidden variables, the explicit manifold increases its volume. By augmenting new feature statistics, the implicit manifold decreases its volume.

4. Experiment on manifold pursuit

4.1. Purpose and results

In this section, we describe an experiment for pursuing the explicit and implicit manifolds by learning from a sample of training image patches. The purpose of this experiment is to illustrate that the two types of manifolds, which correspond to two different classes of models, can be pursued together in the same framework, which gives us a single *mixed* sequence of two types of manifolds.



(a) image (b) sketch

Figure 3. Two of the 75 training images and their sketches used for experiment. Image patches of structures and textures are taken from these images as training examples.

We shall first describe the results before getting into details. The training image patches are taken from 75 images like the two displayed in Fig. 3. The 20 manifolds that are pursued by our method are shown in Fig. 4 in the order of their selections. We can see that the first three manifolds are implicit manifolds of textures, then the explicit manifold of edges is selected. After that the two types of manifolds are selected in mixed order. Fig. 5 shows the frequencies f_k and information gains l_k of the sequentially selected manifolds. The information gains measure the significance of these manifolds, therefore providing a statistical justification for the corresponding two types of models.

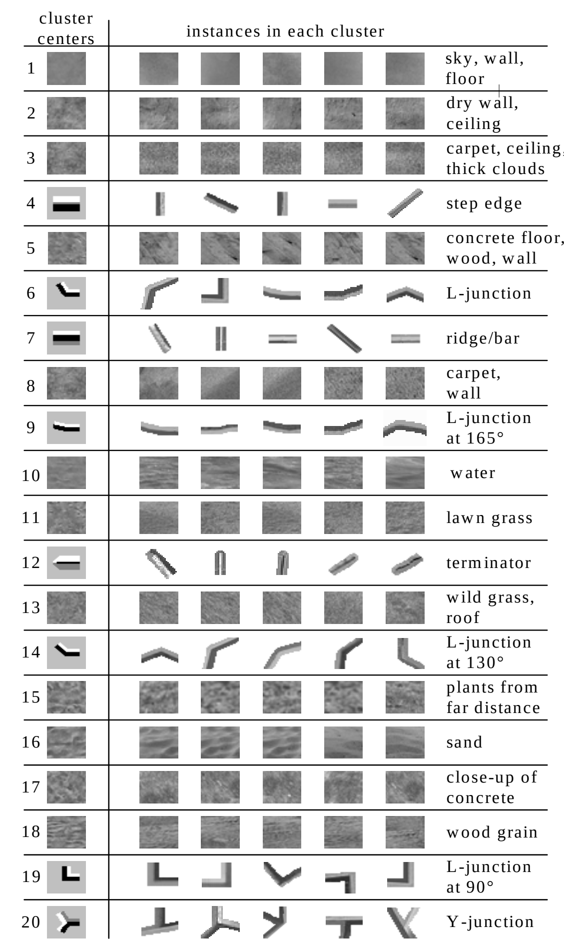


Figure 4. The prototypes of the manifolds sequentially selected, and the instances of image patches on these manifolds. The two types of manifolds are selected in mixed order.

4.2. Details

The training image patches are taken from 75 images, consisting of indoor scenes such as meeting room, bedroom, bathroom, etc., and outdoor scenes such as buildings, mountains, farms, etc. These images are manually sketched and

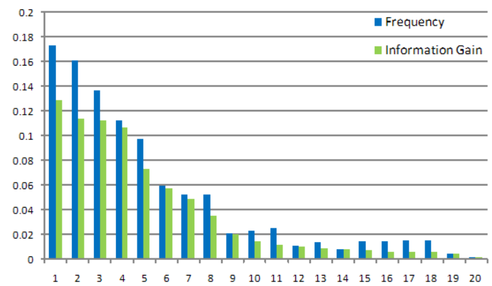


Figure 5. Frequencies and information gains of the 20 sequentially selected manifolds.

labeled by artists. Two examples are shown in Fig. 3. The sketch of an image divides the image domain into structure areas that are around the sketched object boundaries, and the texture areas that do not overlap with the sketched boundaries. For each image, the image patches are taken in such a way that the patches pave the whole image without overlap.

The structured areas of the images are represented by primitive functions. The two most basic primitives are edges and bars. We use six parameters: $(u_1, u_2, w_1, w_{12}, w_2, \theta)$ to represent the edges, which respectively denote the left intensity, the right intensity, the width of the left intensity, the blurring scale over the step transition, the width of the right intensity, and the angle of rotation. Similarly, for bars, we use nine parameters: $(u_1, u_2, u_3, w_1, w_{12}, w_2, w_{23}, w_3, \theta)$, where (u_1, u_2, u_3) and (w_1, w_2, w_3) denote the intensities and widths of the three segments of the profile respectively, (w_{12}, w_{23}) denote the two blurring scales, and θ denotes the angle of rotation.

For each of the labeled sketches, we collect all pixels within 3 pixels of the sketch, creating a collection of 7 pixel wide line segments. These segments are then cut into a sample of 7×11 patches. The intensities of all the image patches are normalized to have mean 0 and variance 1, and the image patches can be rotated into a prototype form such that they would lie horizontally and that the average intensity value of the top half of the patch is less than that of the bottom half. From there, all the patches would be clustered into either edge manifold or bar manifold.

More complex primitives can be built on top of these two simple primitives. Ordering by the degree of connectivity, they are terminators, L-junctions, Y-junctions and crosses. These primitives are compositions of one or more edges/bars, and we may represent them by the combined parameters of the constituent edges/bars. A terminator is simply a bar that is connected to only one other edge or bar, thus no simplification can be made for it. But for the other three primitive types, we do not necessarily need all of these parameters to code them. For example, an L-junction is almost always made up of either two edges or two bars, but almost never an edge and a bar. In addition, the two edges or two bars that make up the L-junction almost always have the same parameter values (except the angles of rotation). Therefore, we can reduce the coding length for most of these L-junctions by nearly one-half. From there, we can formulate two L-junction manifolds, edge/edge and bar/bar. The same clustering procedure is also applied to Y-junctions and crosses. Fig. 4 shows some examples on the explicit manifolds, including the prototypes and the instances on the manifolds.

The textured areas of the images are represented by histograms of filter responses. Our filter bank consists of 17 filters (3 Laplician of Gaussian, 2 Gradient and 12 Gabor), none of the filters are bigger than 7×7. We segment the textured areas of each image into several irregularly shaped regions, usually between 4 to 8 large regions are needed for each image, plus a number of relatively small regions.

The intensities of each region is normalized to have mean 0 and variance 1, and they are represented by a group of histograms. We collect the histograms for all regions in all images, and cluster the regions by the following method.

1. Select the histogram $\mathbf{h}$ that has the largest variance. If the variance is greater than a pre-defined value ϵ, then go to step 2. Otherwise, go to step 4.

2. Cluster $\mathbf{h}$ using the k-means method, k is selected by choosing the smallest value such that the variance of $\mathbf{h}$ within every cluster is smaller than ϵ.

3. For each cluster created, repeat step 1 within the cluster.

4. Terminate.

Each cluster is an implicit manifold. Fig. 4 shows some examples of the implicit manifolds.

Here we assume that the manifolds are non-overlapping, which is approximately true. Then we can select the manifolds sequentially according to the information gain defined by (11), (12), (13).

5. Scale and manifolds

Image patches appear at different scales and resolutions. In this section, we study the effects of scale on the competition between manifolds, as well as on the complexity of the fitted manifolds.

5.1. Competition between manifolds

In the previous section, the structured patches and the textured patches are manually separated out for learning. For an image patch $\mathbf{I}$, it can belong to both an explicit manifold Ω^{ex}_k or an implicit manifold $\Omega^{\text{im}}_{k'}$. The competition between these two manifolds can be automatically carried out by comparing $\log |\Omega_k^{\text{ex}}|$ and $\log |\Omega_{k'}^{\text{im}}|$, which measure the coding lengths by the two manifolds respectively.

Such a competition depends crucially on the scale or resolution, which is an important physical dimension in the ensemble of natural image patches. Image patches at different resolutions can have different entropies, and they should be coded by different manifolds.

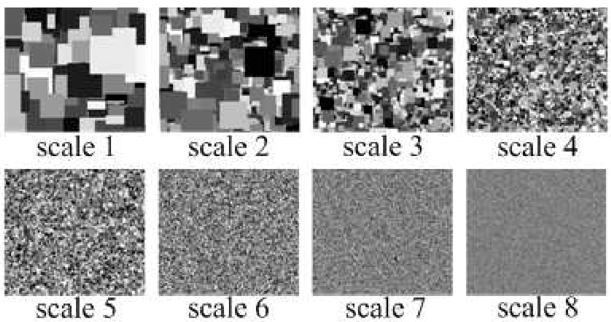


Figure 6. A sequence of images of occluding squares. The resolution of each image is 1/2 of the previous image.

We conduct an experiment to compare the coding efficiencies of the two types of manifolds at different scales. The data we use are nine 512×512 images composed of many occluding squares, shown in Fig. 6. In the first scale, the length of the squares $r \in [64, 256]$, and the the frequency distribution of the sizes is proportional to $1/r^3$. Each subsequent scale is a zoom-out version where the resolution is lowered by 1/2 from the previous image. The intensity of each pixel (x, y) is the average of the four pixels $(2x-1, 2y-1), (2x-1, 2y), (2x, 2y-1), (2x, 2y)$ from the previous scale. All nine images are then normalized to have the same marginal mean and variance.

We compare the coding efficiency of the two manifolds. For each scale, we code the image by a linear sparse coding model $\mathbf{I} = \sum_{i=1}^{d} w_i B_i$, where the image bases B_i are selected from a bank of bases that consists of Haar and Gabor bases by the matching pursuit algorithm [10]. The coding length is computed according to (13). We also code the same image by the implicit manifold based on their feature statistics. The coding length is computed according to (12).

The coding length for the two manifolds are plotted in Fig. 8. We can see that the coding lengths of both coding methods increase as the scale increases, because the images become more complex with higher entropy. But it is clearly more efficient to use explicit manifold to code the high resolution images, and use the implicit manifold to code the low resolution ones. The two curves intersect between scales 4 and 5, indicating that the coding efficiencies of the two manifolds are roughly equal for images at the medium resolution or medium entropy.

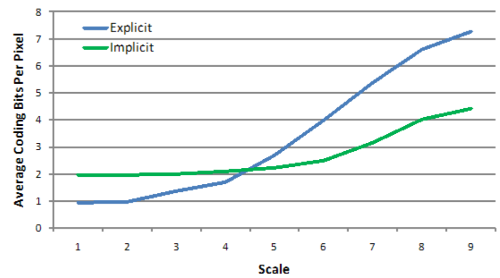


Figure 7. Coding length versus scale.

5.2. Complexity of fitted manifolds

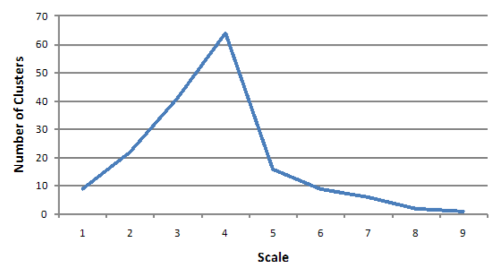


Figure 8. Number of clusters versus scale.

Although the complexity of the image data increases over scale, the complexity of the best fitting manifolds peak at medium resolution, which is the most informative resolution. This can be illustrated by the following experiment.

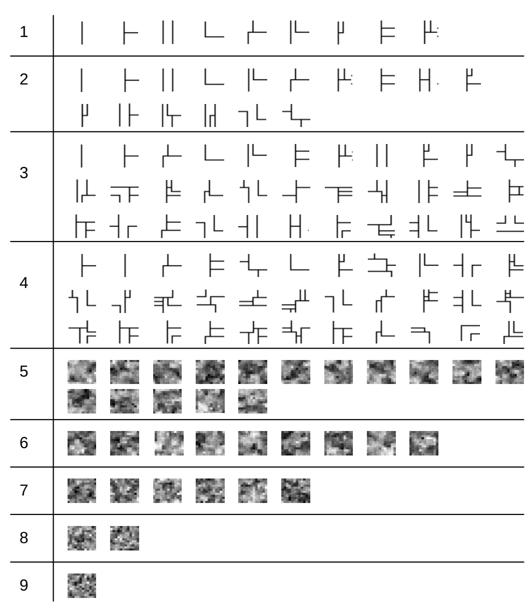


Figure 9. Centers of clusters over scale.

In this experiment, we estimate the number of clusters that can be reliably detected at each scale. From the previous experiment, we see that it is more efficient to code images in scales 1-4 by explicit methods, and more efficient to code images in scales 5-9 by implicit methods. Therefore, we try to identify the number of explicit clusters in the first 4 scales, and the number of implicit clusters in the latter 5 scales.

For explicit clustering, we sketch all the visible borders of the squares of the first scale. For each of the subsequent 3 scales, we generate their sketches by scaling down the labeled sketches from the first scale. If two line segments in the sketch become too close (within 2 pixels), we would allow the two lines to merge into one. Then for each of the four sketches, we randomly select 400 12×12 image patches from each scale and cluster them based on the following 9 parameters: number of L-junctions, number of T-junctions, number of crosses, number of non-intersecting sketches, number of disjoint regions, and number of outgoing lines at each of the four sides of the patches. The clusters with frequency greater than 0.5% are included.

For implicit clustering, we also randomly collected 400 12×12 image patches from each scale, but instead of using the original images, the clustering is done on the histograms of filter responses using the same method described above.

A plot of the number of clusters identified in each scale is shown in Fig.8. The centers of the clusters over scale are shown in Fig.9. We can see that there are only a few clusters at the two ends of the scale range, and the curve peaks at scale 4. This means we only need very simple manifolds to code very high or very low resolution images, but we need more complex manifolds to code images of medium resolution. This suggests that the medium resolution is most informative for object recognition. Over the whole range of resolution, the image patches change from simple regularity to complex regularity to complex randomness to simple randomness.

6. Conclusion

The contribution of this paper is to propose a method for pursuing two different types of manifolds, which give rise to two different classes of models. Moreover, we have examined the relationship between models and scale.

Acknowledgement

This work is supported by NSF grant 0413214 and ONR grant N-00014-05-1-0543. The data used in this paper are provided by the Lotus Hill Annotation project [14], which is supported partially by a sub-award from the W. M. Keck foundation, a Microsoft gift, and 863 grant 2006AA01Z121.

References

[1] S. Della Pietra, V. Della Pietra, and J. Lafferty, “Inducing features of random fields,” *IEEE-PAMI*, 19(4), 380–393, 1997. 3

[2] J.H. Elder and S. W. Zucker, “Local scale control for edge detection and blur estimation,” *IEEE-PAMI*, 20(7), 699-716, 1998. 2

[3] D. Geman and A. Koloydenko, “Invariant statistics and coding of natural microimages,” *2nd Int’l Workshop on Stat. and Comp. Theory of Vision* (SCTV), 1999. 1

[4] C. E. Guo, S. C. Zhu, and Y. N. Wu (2006) “Primal sketch: integrating structure and texture,” *Computer Vision and Image Understanding*, 106, 5-19. 2

[5] F. Jurie and B. Triggs, “Creating efficient codebooks for visual recognition,” *ICCV05*, I: 604-610, 2005. 1

[6] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: spatial pyramid matching for recognizing natural scene categories,” *CVPR06*, II: 2169-2178, 2006. 1

[7] A. B. Lee, K. S. Pedersen and D. Mumford, “The nonlinear statistics of high-contrast patches in natural images,” *IJCV*, 54 (1 / 2): 83-103, 2003 1, 2

[8] F. F. Li and P. Perona, “A Bayesian hierarchical model for learning natural scene categories,” *CVPR05*, II: 524-531, 2005. 1

[9] Y. Ma, H. Derkesen, W. Hong, and J. Wright, “Segmentation of multivariate mixed data via lossy coding and compression,” *IEEE-PAMI*, (to appear), 2007. 2

[10] S. Mallat and Z. Zhang, “Matching pursuit in a time-frequency dictionary,” *IEEE Signal Processing*, 41, 3397-415, 1993. 6

[11] D. Mumford and J. Shah, “Optimal approximations by piecewise smooth functions and associated variational problems,” *Comm. Pure Applied. Math.*, 42, 577-685, 1989. 2

[12] B. A. Olshausen and D. J. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, 381, 607-609, 1996. 1

[13] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, “Discovering objects and their localization in images,” *ICCV05*, I: 370-377, 2005. 1

[14] B. Yao, X. Yang, and S. C. Zhu, “Introduction to a large-scale general purpose ground truth database: methodology, annotation tool and benchmarks,” *EMMCVPR07*, (to appear), 2007. 7

[15] S. C. Zhu, Y. N. Wu, and D. Mumford, “Minimax entropy principle and its applications in texture modeling,” *Neural Computation*, 9(8), 1627-1660, 1997. 1, 2