Structure v.s. Appearance and 3D v.s. 2D? A Numeric Answer

Wenze Hu, Zhangzhang Si, and Song-Chun Zhu

1 Introduction

It has been widely acknowledged that while humans are quite good at extracting structures from images, i.e. the edges [4], textons [10] etc. which are concepts hidden in pixel intensities, the notion of structure does not lend itself to its precise detection by computer programs. As a result, there now exist appearance based image representations [14, 6] which directly express the image using statistics or histograms of image operator (filter) responses. Structure based and appearance based image representations are advocated by different researchers, whose reasons for endorsement range from the practical benefits in building simple vision applications to the faith that computer vision would ultimately stick to human vision.

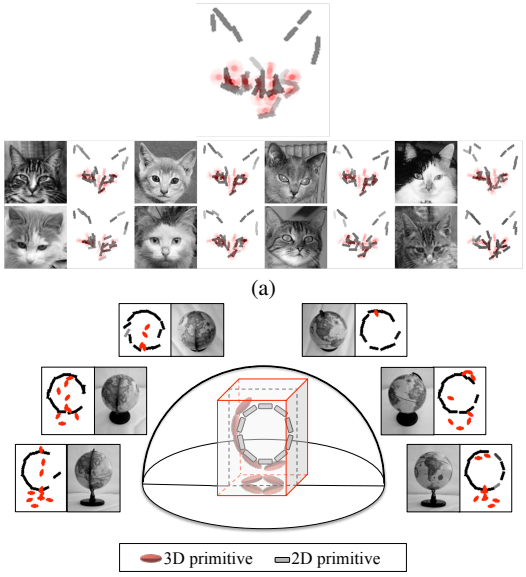
When different views of objects are taken into account, a similar dichotomy happens in describing the image structures. The intrinsic 3D shape of objects suggests that object-centered representation using volumetric primitives [2, 1, 15] should be simple yet capable of representing the observed image structure changes. But again the difficulty of extracting these 3D hidden concepts from images make the viewer-centered representation [11, 12, 17, 21] a competing alternative, which uses a collections of 2D representations each covering a small portion of the modelled views. Over the two representations, researchers showed various cases where one representation prevailed [20, 3, 8], but there is no clear winner.

In our view, these competing representations are points lying in different positions of representation spectrum, and they should be combined to better represent images. For example, consider the images of leaves at different scales shown in Fig.1, one can easily identify the structures inside the first image, but quickly give up and change to appearance based description for the last image. By gradually zooming the camera, images in between must combine some portion of both struc-

Wenze Hu, Zhangzhang Si and Song-Chun Zhu
University of California, Los Angeles, CA USA. e-mail: {wzhu,zzsi,sczhu}@stat.ucla.edu

Fig. 1 Images of leaves at different scales. From left to right, our description of the image gradually changes from exact structure to the overall appearance of the leaves.

ture and appearance. A similar spectrum for the 2D and 3D case is suggested in [7].



(a)

(b)

Fig. 2 (a) Hybrid image template mixing sketch (dark bar) and texture (red disk) representations. (b) Templates mixing 2D primitives (dark bar) and 3D primitives (red bar) to describe the desktop globe images over different views. Image in (b) is adopted from [9].

In this chapter, we want to evaluate and combine these representations on purely intrinsic and quantitative measurements. The idea behind this work is that elements (primitives) in competing representations should be weighted by their information contribution. Borrowing from information theory, we take information gain as a quantitative measure of this contribution. We further introduce a mathematical framework called information projection, which evaluates and sequentially selects elements from both competing representations, so that the best representation of a given set of images can be learned automatically.

As a result of using a combined pool of representational elements, we find that the learned result almost always mixes the competing representations. Fig.2 shows

two typical examples for the structure v.s. appearance case and 3D v.s. 2D case in our study. In the structure v.s. appearance case, we take deformable Gabor filters [16] as primitives for structure representation, and oriented histograms of Gabor responses as those for appearance. On cat face images shown in Fig.2(a), Gabor filters (dark bars) are automatically selected to describe the boundary of cat ears, eyes and mouth etc., while the histograms (red disks) are used to encode the texture of fur on the cat face. Similarly, in the 2D v.s. 3D case, we take 2D Gabor filters and 3D stick-like elements as primitives for object-centered representation and viewer-centered representation respectively. Given the set of desktop globe images shown in Fig.2 (b), the algorithm selects 3D primitives (red bars) for the handle and the base as their appearance change drastically across views, and uses 2D primitives (dark bars) for the much more view invariant circular shape of the globe. More experiments on various image classes reveal that the representation spectrum exist, which further shows the importance of having a numerical solution to the representation integration problem.

In the rest of this chapter, we will first introduce the information projection framework along with our information gain criterion, followed by detailed case studies over the above two pairs of competing representations.

2 Information Projection

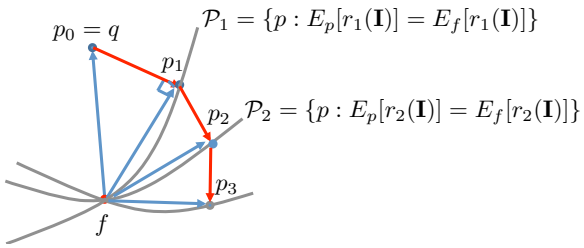


Fig. 3 Learning by information projection, illustrated in the model space. Each point in this space represents a possible model or a distribution over the target images. The series of models $p_0, p_1, \ldots, p_K$ converge to the target probability f monotonically by sequentially matching the constraints. Image adopted from [19].

We treat the set of images we want to model as samples from a target image distribution $f(\mathbf{I})$:

$$\{\mathbf{I}_m^{obs}; m = 1, 2, \cdots, M\} \sim f(\mathbf{I}), \tag{1}$$

Our objective is to learn a sequence of models p starting from an initial reference model q, which would incrementally approach the target distribution f by minimizing the Kullback-Leibler divergence $KL(f||p)$:

$$q = p_0 \rightarrow p_1 \rightarrow \cdots p_k \text{ to } f \tag{2}$$

The approach can be explained in the space of probability distributions shown in Fig.3, where each point is a model describing a probability distribution over target images. Our true model f and the initial model q are two points in the space with a large divergence $KL(f||q)$.

The learning proceeds iteratively. In each iteration, we augment the current model p_{k-1} to p_k, by adding one statistical constraint so that p_k matches a new marginal statistics $E_f[r_k(\mathbf{I})]$, where $r_k(\mathbf{I})$ is a scalar or vector valued function of images, denoting the response of elements (such as Gabor filter) in image representations. Specifically, an iteration is composed of the following two steps:

1. *Min-step*. Given an unobserved image statistics $E_f[r_k(\mathbf{I})]$, we want to find the model p_k^* in the set of models $\mathscr{P}_k$ having the same statistics as f while as close to p_{k-1} as possible:

$$\mathscr{P}_k = \left\{ p : E_p[r_k(\mathbf{I})] = E_f[r_k(\mathbf{I})] \right\}. \tag{3}$$

The closeness is evaluated by $KL(p_k||p_{k-1})$. Intuitively, this is to enforce that on the element k our model should produce the same image statistics as the true model f. In figure.3, this set of models $\mathscr{P}_k$ can be shown as the corresponding gray curve passing through f. According to the pythagorean theorem [5] in information theory, the new model p^* is the perpendicular projection of p_{k-1} on $\mathscr{P}_k$, and the three points f, p_k^*, p_{k-1} form a right triangle.

The step above actually solves the constrained optimization problem of

$$p_k^* = \underset{p \in \mathscr{P}_k}{\operatorname{argmin}} KL(p||p_{k-1}). \tag{4}$$

By using Lagrange multiplier, we have

$$p_k(\mathbf{I};\Theta_k) = \frac{1}{z_k} p_{k-1}(\mathbf{I};\Theta_{k-1}) e^{-\lambda_k r_k(\mathbf{I})} \tag{5}$$

where λ_k is the parameter satisfying the constraint in Eqn.3, z_k is the partition function that normalize the probability to 1, and $\Theta_k = \{\lambda_1, \lambda_2, \cdots, \lambda_k\}$.

2. *Max-step*. Among all the candidate elements $r(\mathbf{I})$ and their statistics, choose the one that reveals the largest difference between p_k and p_{k-1}.

$$r_k^* = \arg\max KL(p_k||p_{k-1}) \tag{6}$$

As the KL-divergence is non-negative and

$$KL(p_k||p_{k-1}) = KL(f||p_{k-1}) - KL(f||p_k), \tag{7}$$

this step greedily minimize the the KL-divergence between f and our final model p. Intuitively, this step chooses a curve in Fig.3 which is farthest away from the current model p_{k-1}.

After K iterations, we obtain a model

$$p(\mathbf{I}|\Theta) = q(\mathbf{I}) \prod_{k=1}^{K} \frac{1}{z_k} e^{-\lambda_k r_k}, \tag{8}$$

and the information gain of each step k is:

$$Ig_k = E_{p_k}\left[\log \frac{p_k(\mathbf{I}|\Theta)}{p_{k-1}(\mathbf{I})}\right] = KL(p_k||p_{k-1}) \tag{9}$$

As the information gain in step k is equal to $KL(p_k||p_{k-1})$, each of the training iterations above actually selects the representation element which achieves maximum information gain over the current model p_{k-1}. This learning process sequentially projects the current model to a number of constrained spaces, and thus is called information projection.

As the only assumption in the above model is that the candidate element should have a scalar or vector valued response, the information projection framework can be used in many feature or pattern learning problems, provided that the goal of learning is to construct the target image distribution, and the candidate element pool is fixed before learning. In the following, we discuss in detail the candidate element pools and implementation details in studying the two groups of competing representations introduced Section 1.

3 Case I: Combining Sketch and Texture

In the first case study, we illustrate the integration of structure and appearance as hybrid image templates (HIT) for object image modelling. More discussion about this hybrid image template can be seen in [19].

We assume the training images $\{\mathbf{I}_m^{obs} : m = 1, 2, \cdots, M\}$ are instances of an object category and are roughly aligned in position, orientation and scale with arbitrary background, such as the ones shown in Fig.2(a). While a single template suffices for the cat examples here, when there is large pose variations on the objects in training images, multiple templates can be learned through an EM-like clustering procedure.

Given the set of training image, the lattice of the image Λ is decomposed into a set of K non-overlapping patches $\{\Lambda_i\}_{i=1}^{K}$ selected from a large pool of candidate patches. Note that these patches do not necessary compose the full lattice Λ as some pixels in Λ may correspond to object background which have inconsistent feature responses.

By enumerating all possible patch candidates, and assuming each patch can be represented either by its structure or appearance, we are able to construct a large pool of candidate representation elements. It is worth noting that although the final selected set of patches are assumed to be non-overlapping, patches in the candidate pool are not subject to this restriction.

Currently, we limit the structure elements to be sketches only, such as those shown on the boundary of the hedgehog template in Fig.4(b). If an image patch

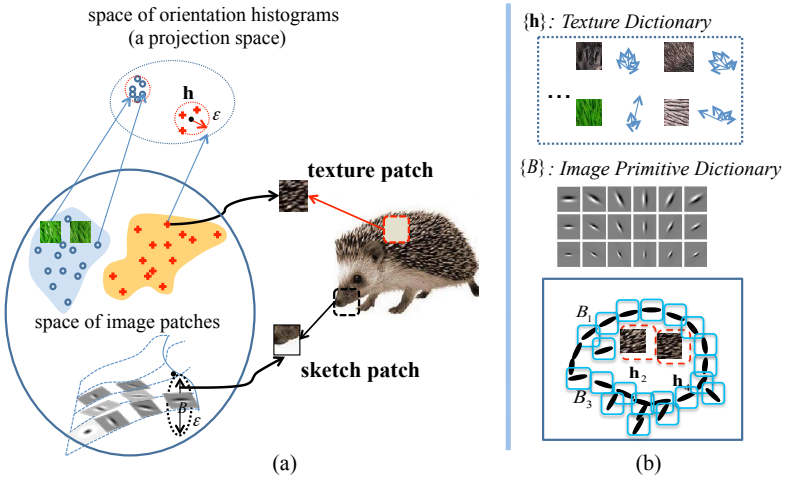


Fig. 4 (a) A hedgehog image may be seen as a bunch of local image patches, being either sketches or textures. (b) Quantization in the image space and histogram feature space provides candidate pools for sketches $\{B\}$ and textures $\{\mathbf{h}\}$ respectively. A hybrid template of hedgehog $T = \{B_1, \mathbf{h}_2, B_3, \mathbf{h}_4, \cdots\}$ is composed of sketches and histogram prototypes explaining local image patches at different locations. Image adopted from [19].

$\mathbf{I}_{\Lambda_k}$ is represented as a sketch, we define its element response $r^{skt}(\mathbf{I}_{\Lambda_k})$ by

$$r^{skt}(\mathbf{I}_{\Lambda_k}) = max_{\mathrm{d}x\in\partial x,\mathrm{d}o\in\partial o} \mathrm{S}(|\langle \mathbf{I}, B_{x_k+\mathrm{d}x,o_k+\mathrm{d}o}\rangle|^2) \tag{10}$$

which is a transformed local maximum response of a image primitive B around a local neighborhood of a specific position x and orientation o indexed by k. Here we choose the primitives to be Gabor filters at a set of discrete orientations, such as the ones shown in Fig.4(b), $\langle\cdot,\cdot\rangle$ denotes the inner product between the image and the filter, and $\mathrm{S}(\cdot)$ is the sigmoid transform that saturates large filter responses.

Similarly, we limit the appearance elements to be those for texture only, and define the element response as:

$$r^{app}(\mathbf{I}_{\Lambda_k}) = \mathrm{S}(\|H(\mathbf{I}_{\Lambda_k}) - \mathbf{h}\|^2) \tag{11}$$

where $H(\mathbf{I}_{\Lambda_k})$ is the histogram of the responses from Gabor filters at different orientations pooled within $\mathbf{I}_{\Lambda_k}$ and $\mathbf{h}$ is a pre-computed histogram prototype (one may consider it as a cluster center of similar texture patches). In practice, $\mathbf{h}$ is obtained by averaging the histograms at the same position over all the observed training examples.

By constructing this candidate pool, we derive a large set of constraints as in Eqn.3 on the individual response of patch representations, where $E_f[r(\mathbf{I}_{\Lambda_k})]$ is estimated by the average response on observed images

$$E_f[r(\mathbf{I}_{\Lambda_k})] \approx \frac{1}{M}\sum_{m=1}^{M} r(\mathbf{I}_{m,\Lambda_k}) \tag{12}$$

Because we assume the selected patches are non-overlapping,

$$Ig_k = KL[p_k(\mathbf{I})||p_{k-1}(\mathbf{I})] = KL[p(\mathbf{I}_{\Lambda_k})||q(\mathbf{I}_{\Lambda_k})], \tag{13}$$

thus the information gain of each candidate element can be computed ahead of time by matching $E_p[r_k(\mathbf{I})] = E_f[r_k(\mathbf{I})]$. During learning, the patch non-overlapping assumption is enforced by inhibiting (removing) candidate patches with significant overlap with selected ones.

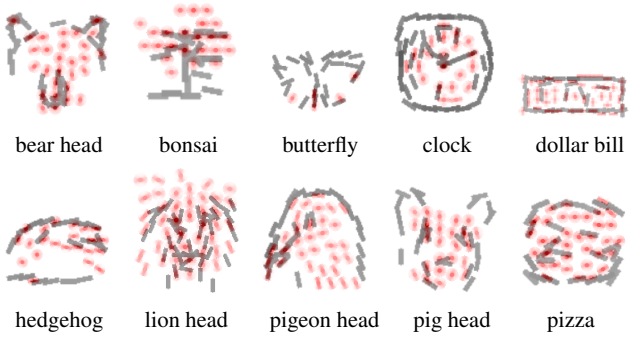


Fig. 5 Learned HITs of several object categories. Bold black bars denote sketches, while red blobs denote texture patches. For illustration purpose, we only show sketches/textures of a single scale and vary the (relative) Gabor scales for different categories.

Fig.5 shows the hybrid templates learned from several categories. The sketches usually outlines the rough shape of the target object category, with the appearance patches fill in the furs on animal head or leaves on the tree.

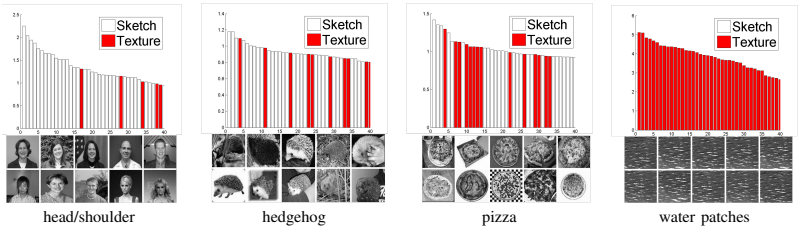


Fig. 6 Transition from structure based to appearance based representation. For each image category, the top 40 selected patches are ordered by their information gains in decreasing order. Image adopted from [19].

We also study how the structure and appearance patches are ordered by their information gains. We choose four categories ranging from structured to textured: head-shoulder, hedgehog, pizza, and wavy water. We plot the information gains of the selected patches in decreasing order: the hollow bars are for structure patches and the solid (red) bars are for texture patches. For image categories with regular shape, e.g. head/shoulder, sketches dominate the information gain. For the wavy water, textures makes the most contributions. The two categories in the middle contains different degrees of mix of sketch and texture.

Learned templates can be used for image classification. Quantitative experiments in [19] show that our method performances on-par with HoG+SVM approach [6] on several public datasets, while using far shorter (at least 1/10) feature dimensions.

4 Case II: Mixing 3D and 2D Primitives

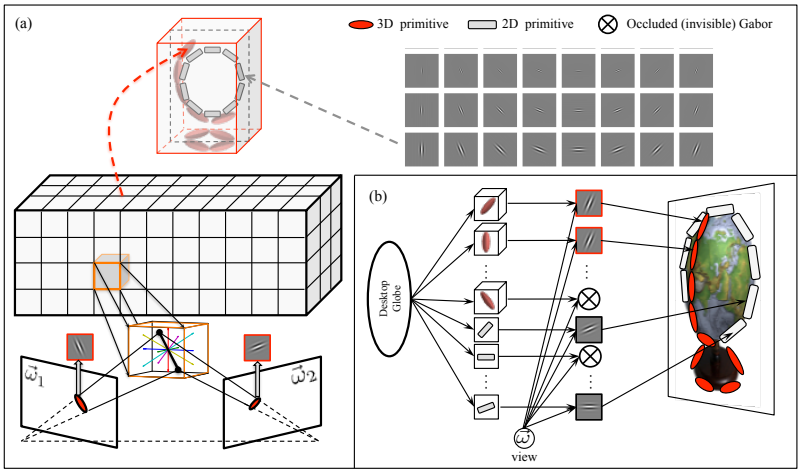


Fig. 7 (a) Illustration of 3D and 2D primitives and how they are used to compose mixed representations. The 3D primitives can be viewed as sticks with selected 3D positions and orientations, while 2D primitive are Gabor filters located at selected 2D positions and orientations. (b) When generating object images at a particular view ω, we project the 3D primitives and superimpose the 2D primitives. A primitive is not instantiated if it is not visible in the particular view. Image adopted from [9]

In the second case study, we illustrate the automatic selection of viewer-centered and object-centered representations, which build object templates composed of 3D and 2D primitives. The observed images are those of an object category captured from different views $\{\mathbf{I}_m, \omega_m\}_{m=1}^{M}$, where ω denotes the view of the image parameterized by the pan and tilt angle of camera. For simplicity, we assume the camera

roll angle is zero, and object images are captured at the same scale. More discussion about this case study can be found in [9].

Similar to the structure v.s. appearance case, we have two types of image primitives to build our pool of constraints.

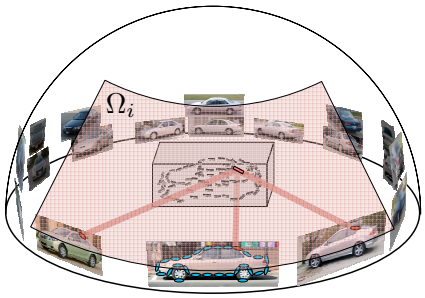


Fig. 8 Illustration of how 3D primitives are learned from images in different views. The learning step can be interpreted as trying all possible locations and orientations of 3D primitives and incrementally select ones with high overall responses. Image from [9]

The first type is the primitive for object-centered representation, which we call 3D primitives. The 3D primitives are stick like primitives with position X and orientation O in 3D space, such as the ones shown in the center of Fig.8. On object images, primitives are realized by Gabor filters at their projected positions and orientations, and primitive responses can be defined similar to the sketch responses in the previous case study:

$$r_k^{3D}(\mathbf{I};\omega) = max_{\mathrm{d}x\in\partial x,\mathrm{d}o\in\partial o}\mathrm{S}(|\langle \mathbf{I}, G_{P(X_k,\omega)+\mathrm{d}x,P(O_k,\omega)+\mathrm{d}o}\rangle|^2) \tag{14}$$

where $P(\cdot,\cdot)$ denotes the camera projection function.

Our 3D primitive pool is created by enumerating combinations of 3D positions X and orientations O as in Fig.7(a). To avoid an excessive enumeration of 3D primitives, we quantize the 3D object volume into non-overlapping cuboids, and inside each cuboid, we sample primitive orientations uniformly.

We illustrate how the proposed 3D primitives pool information from images in Fig.8. For each hypothesized 3D primitive, we project it on to observed images from different views, compute its primitive responses on images and estimate its statistics. Primitive responses of meaningful ones will be consistently high across views and will contribute significant information gains.

The 2D primitives we choose for viewer-based image representation are the same as the sketches in case study I, and thus are not further explained.

Compared with learning the hybrid image templates model in section 3 , one key difference in mixing 3D and 2D primitives is that the primitives might occlude each other, such as in the desktop globe example. Also, as a viewer-centered representation, the 2D primitives should be allowed to be view specific, thus individual

2D sketches may only explain some of the observed images whose views are in a specific range.

To model these effects, we introduce another auxiliary variable Ω to describe the visible range of a primitive, which is defined as a set of N views on which the corresponding primitive is visible: f$\Omega = \{\omega_1, \omega_2, \cdots, \omega_N\}$

Adding this auxiliary variable leads to the following changes in *Min-step* and *Max-step*:

In *Min-step*, the estimation of $E_f[r(\mathbf{I})]$ is changed to the sample mean of the primitive response on visible images:

$$E_f[r(\mathbf{I})] \approx \frac{1}{N} \sum_{m=1}^{M} r_k(\mathbf{I}_m) \cdot \mathbf{1}(\omega_m \in \Omega_k) \tag{15}$$

where $\mathbf{1}(\cdot)$ is an indicator function that equals 1 if ω_m is in set Ω_k, and 0 otherwise. In Eqn.(15), $N = \sum_{m=1}^{M} \mathbf{1}(\omega_m \in \Omega_k)$.

In *Max-step*, the constraint selection criterion is also updated to

$$(r_k^*, \Omega_k^*) = \arg\max KL(p_k || p_{k-1}) \tag{16}$$

Conceptually, adding auxiliary variable Ω will dramatically increase the computational complexity, since for each primitive, we need to search through 2^M possible different view ranges. However, since we only care about the pair which leads to maximum information gain, these 2^M evaluations would be reduced to less than M evaluations. Details of this simplification can be found in [9].

For experiments, we use the ETH80 dataset [13], and further augment it by adding images of soda cans and desktop globes to have 10 categories of object images, where each one is captured from different views and with a few instances. Due to the difficulty in illustrating the 3D elements in mixed templates, we directly show their projections on sample training images in the first three rows of Fig.9.

From the figure, we can see that our method automatically finds suitable representations for different object categories, which spans a spectrum from nearly pure 2D to pure 3D. For object categories with stable 2D shapes, the algorithm automatically selects 2D primitives to form the rough shape of that category. For parts such as the tip of tomatoes, and handle of cups, it selects 3D primitives, because these details are view specific and only appear in part of the view sphere. For categories with complex shapes, coding the projected shape for each single view will be less efficient than coding the general 3D shape using 3D primitives. So, the algorithm automatically transit to mostly selecting 3D primitives.

Row 4 and 5 of Fig.9 shows the selection order of the 3D (red) and 2D (gray) primitives, their information gains, and their proportion of information contribution in each template. By sorting these categories according to this proportion, Fig.9 clearly shows that the learned representations reside in different positions of the spectrum, and their positions are related to the complexity of the object shape.

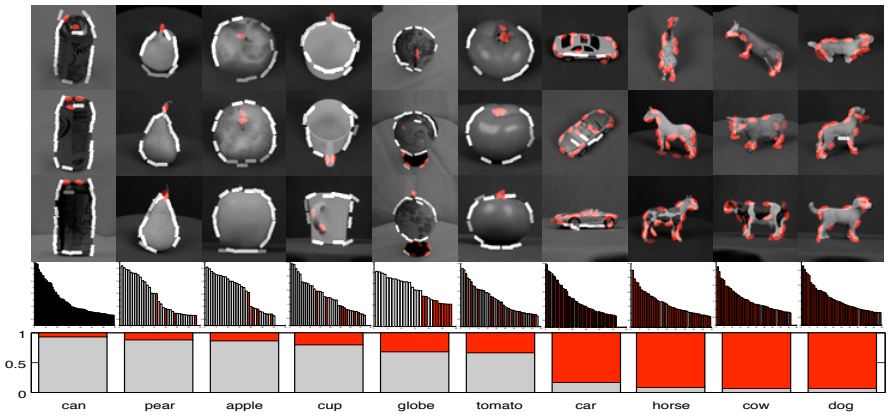


Fig. 9 Spectrum of image representations in 2D/3D case. **Row 1-3**: Learned templates for each object category. The white bars represent 2D primitives and red ellipses are 3D primitives. **Row 4**: The selection order of 2D and 3D primitives in each object class and their information gains. **Row 5**: The ratio on the information contribution of 3D and 2D primitives. Image from [9]

Fig. 10 The confusion matrix over 8 poses in the 3D car dataset [18], and sample car pose estimation results. 3D and 2D primitives are all showed using white bars. Image adopted from [9].

The learned model can be used for image classification and object pose estimation. Fig.10 shows sample pose estimation results and the confusion matrix over the 8 views in 3D car dataset [18]. More experiment results can be found in [9].

5 Discussion

This chapter introduces a general learning framework that automatically mixes different representations to find the best representation for a given set of images. By using images of different object classes, we show that there are representation spectrum where images are best represented by mixing different proportions of competing representations. Although only particular cases are illustrated, the framework we describe permits the exploration of this approach to general representations and we hope it will prove to be useful for researchers in the vision community.

Acknowledgements This work is supported by DARPA grant FA 8650-11-1-7149, NSF IIS1018751 and MURI grant ONR N00014-10-1-0933.

References

1. Barr, A.: Superquadrics and angle-preserving transformations. IEEE Computer Graphics and Applications **1**(1), 11 –23 (1981)
2. Biederman, I.: Recognition-by-components: A theory of human image understanding. Psychological Review **94**, 115–117 (1987)
3. Biederman, I., Gerhardstein, P.C.: Viewpoint-dependent mechanisms in visual object recognition: Reply to tarr and bülthoff (1995). Journal of Experimental Psychology: Human Perception and Performance **21**(6), 1506–1514 (1995)
4. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**, 679–698 (1986)
5. Csiszár, I., Shields, P.C.: Information theory and statistics: A tutorial. Now Publishers Inc (2004)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, pp. 886–893 (2005)
7. Dickinson, S., Pentland, A., Rosenfeld, A.: From volumes to views: an approach to 3-d object recognition. In: Workshop on Directions in Automated CAD-Based Vision,, pp. 85 –96 (1991)
8. Hayward, W.G., Tarr, M.J.: Testing conditions for viewpoint invariance in object recognition. Journal of Experimental Psychology: Human Perception and Performance **23**(5), 1511–1521 (1997)
9. Hu, W., Zhu, S.C.: Learning a probabilistic model mixing 3D and 2D primitives for view invariant object recognition. In: Computer Vision and Pattern Recognition, pp. 2273–2280 (2010)
10. Julesz, B.: Textons, the elements of texture perception, and their interactions. Nature **290**(5802), 91–97 (1981)
11. Koenderink, J., Doorn, A.: The singularities of the visual mapping. Biological cybernetics **24**(1), 51–59 (1976)
12. Koenderink, J., Doorn, A.: The internal representation of solid shape with respect to vision. Biological cybernetics **32**(4), 211–216 (1979)
13. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: CVPR (2003)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004)
15. Marr, D.: Vision. A Computational Investigation into the Human Representation and Processing of Visual Information. W.H. Freeman and Company (1982)
16. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature **381**, 607–609 (1996)
17. Poggio, T., Edelman, S.: A network that learns to recognize three-dimensional objects. Nature **343**(6255), 263–266 (1990)
18. Savarese, S., Fei-Fei, L.: 3d generic object categorization, localization and pose estimation. In: ICCV (2007)
19. Si, Z., Zhu, S.C.: Learning hybrid image templates (hit) by information projection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1354–1367 (2012)
20. Tarr, M.J., Bülthoff, H.H.: Is human object recognition better described by geon structural descriptions or by multiple views? comment on biederman and gerhardstein (1993). Journal of Experimental Psychology: Human Perception and Performance **21**(6), 1494–1505 (1995)
21. Ullman, S., Basri, R.: Recognition by linear combinations of models. IEEE Trans. Pattern Anal. Mach. Intell. **13**(10), 992–1006 (1991)