A Structural Translation Image Representation

April 4, 2012 Bill Freeman, CSAIL, MIT
this file: notes/transRep.tex

1 An Image Representation for Machine Learning

When we apply statistical and mathematical operations to images, we often assume that the images will behave like *vectors* when they are combined. If A and B are both images, we want (A+B)/2 to be a meaningful image, in some way intermediate to both A and B. This assumption about the behavior of images is pervasive in machine learning, computer vision, and computer graphics. It is an unstated assumption behind many of the dominant operations in those fields: principle components analysis, multi-linear models, support vector machines, and many of the generative and discriminative statistical models applied in image analysis and synthetis.

However, if we represent images as pixels, that basic assumption–that images combine like vectors–is in general not met. If we average two images, the result, a "double exposure" image containing both input images at half contrast, is not intermediate to the two input images in any meaningful perceptual way. The subsequent algorithms need to introduce many steps to overcome the limitations caused by the failure of the initial assumption to hold.

Our proposed research program is to develop a vector space representation for images. The representation would exploit what we currently know about image structural similarity in order to allow intermediate images to contain image structures at intermediate locations. We plan to design the image representation in an "open architecture", allowing the representation to be improved over time as more is learned about mathematical representations of perceptually meaningful image constructs.

1.1 An image representation based on structural displacements

First, we describe a displacement-based representation for black-and-white images which gives some hope for finding a representation that will allow general images to combine as vectors do.

The images in Figure 1 show a simple case of the main idea. The average of a pixel representation of the two dot images gives a "double exposure" of each dot (bottom row, left), while an average of a positional representation of the two dots gives a more sensible result: a dot at the average position of each of the input dots.

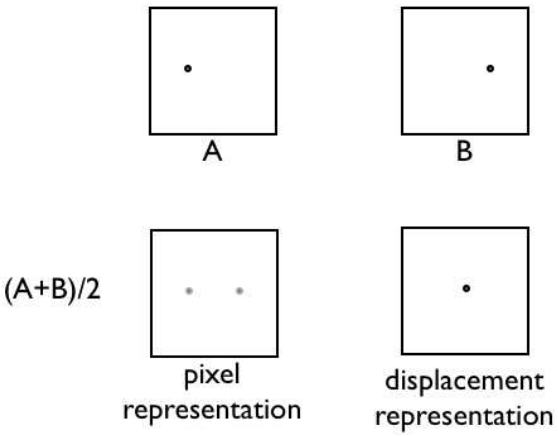
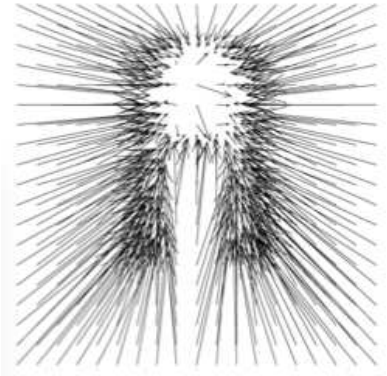
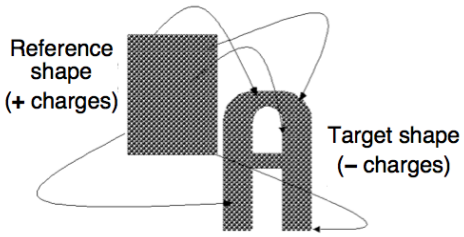


Figure 1: Simplest possible images, showing the main idea. Top row: Images A and B. Bottom row: their vector average, in both a pixel representation, and in a displacement-based representation. In a displacement-based representation, the *location* of the features are stored. When two images are averaged, a dot is displayed at the average location of the two input dots. Note that the displacment-based representation causes the images to behave in an intuitive way.

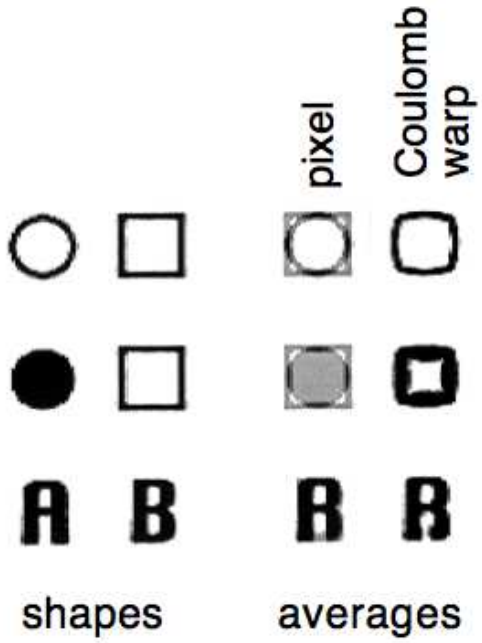
Image representations have been made using that principle. One such representation is the “Coulomb warp” representation, in [4]. An arbitrary black-and-white shape is represented by the set of displacements that a rectangle of ink particles would have to undergo in order to create the desired shape. This representation has good performance in generating sensible intermediate shapes between averaged images, as shown in Figure 2. The key to this representation’s performance is in finding a good assignment strategy for which ink particle of the reference shape will match to which region of the shape to be representated by the translated ink particles. An electrostatic model was used to make that assignment, hence the name “Coulomb warp”.



(a)



(b)



(c)

Figure 2: Coulomb warp representation [4]. (a) (b) (c)

But the Coulomb warp representation only applies to two-tone images. We desire a generalization of it that will apply to high-resolution, full-color images. The desired behavior of our image representation is shown in Figure 3, from [3], made using geometric reasoning. The Mona Lisa, and it’s left-right reflection, were combined, using view morphing, to create a Mona Lisa at a straight-on pose. We would like to be able to create a similar result by (a) encoding the left and right images into our desired representation, averaging them, and then (b) output that back to a pixel representation to display something like the middle image shown here. We want something in the spirit of the Blanz and Vetter work [1], but which applies to all possible images.

We can measure structural similarites using SIFT features, much as was done in the SIFT flow paper by Liu et al

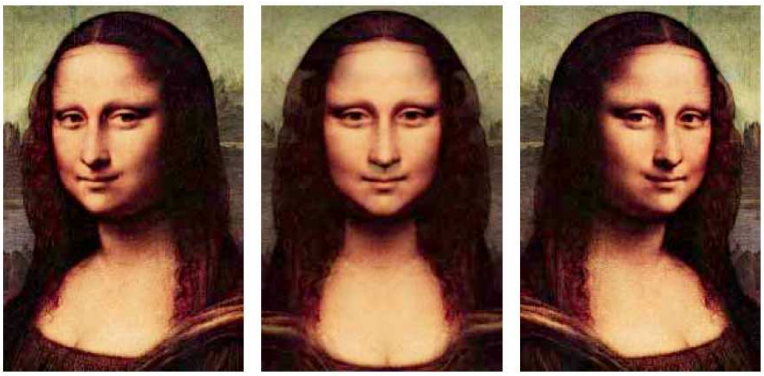


Figure 3: **The goal**. Image by Seitz and Dyer [3], constructed using view morphing. A desired image representation would be one that rendered the average of the two edge images to be something like the middle picture.

[2].

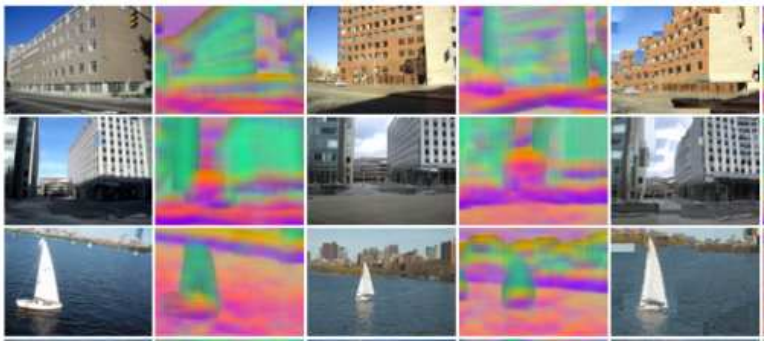


Figure 4: Sift Flow [2] has been used to warp one image to another based on structural similarities.

1.1.1 The approach

Our approach will be:

- To create a structural vocabulary of several thousand sift features–visual words.
- To record the position and value of all sift features–visual words–in an input image. The image representation would be a sparse set of vector positions for where each sift feature is in the image. When there is more than one visual word in a given image, we need to represent those multiple locations in some orderless way. We also need to assign entries to bins in a way that doesn't suffer from histogram boundaries. Probably we'll need to "blur" the histogram entries into neighboring SIFT words.
- To combine entries, combine them as you would any vectors.
- To render the pixel image corresponding to a given sparse set of visual words, we'll need to solve an optimization problem: find the most probable pixel image consistent with the specified collection of visual words at the specified positions. We'll have to figure out how to include color, as well, although one could imagine several possible ways to do that.

The output of this research project would be two functions, **im2vec** and **vec2im**.

im2vec would encode a full-color image as a collection of spatial displacements of SIFT vectors. Color is encoded, too. Linear combinations of encoded images would yield sensible output images–structures with similar SIFT features would appear in the averaged output image at the average of the spatial positions of the input images, and with a structural SIFT feature that was the average of the input values for that feature.

vec2im would decode to a pixel representation. This would involve solving an optimization problem to find the pixel image that best respected the specified SIFT features at the specified image locations.

Figure 5: From the paper, "Reconstructing an image from its local descriptors", [5]. This is kind of an existence proof that images can be reconstructed from SIFT features and their positions. An alternative approach: epitomes. We'd want higher quality image reconstructions those depicted here.

2 More thoughts

A unit normalized version of this vector would be just a bag-of-words representation of the image.

What happens when you project one vector image A onto a unit vector representation of the other one, B? Do you place the things of B at the positions of A? Or maybe you show what image things of image A are similar structurally to image B?

It would be nice to structure the representation in an extensible way

Of course, it should be structured to allow for both short vector representations, with crappy looking images, as well as long vectors, with perfect fidelity: im = vec2im(im2vec(im)).

2.1 Desired characteristics

- the average of two images with spatially offset objects is an image with those objects at their average positions.
- An average over scale translations gives the object at an intermediate scale.
- For the extended, future version: as more and more image interpretation is coded into the representation, those coded interpretations should behave in a vector way, too. Lighting, shadows, occluding boundaries, should all average in the appropriate way, provided the image is encoded properly.

3 SIFT generalization of Coulomb warp map

The canonical shape for the Coulomb warp was a rectangle of ink particles. The canonical shape for SIFT charge representation is an average or sum or vector quantization of SIFT descriptors and their positions over many many images. Free parameter count: number of spatial scales over image. Number of images used for the average reference image. Number of sift features, N, in the vocabulary.

Here's the key algorithm: for each of the N reference sift features, we launch it toward the negative charges.

what the coulomb warp stuff did well: ensured uniform coverage over the target letter shape.

each sift feature has a contrast, and a position. and all of them have a descriptor and scale tuning sigma.

goals:

uniform coverage over the image

3.1 optimization criteria for tuning the free parameters

- reconstruct back to original image.

- intermediate shape gives intermediate representation.
- average of representations gives perceptually intermediate image.
- intermediate scale gives intermediate representations
- some natural path for information about image rendering to improve the vector representation
-

design elements

- local feature
  - should allow reconstruction back to an image, with photo quality.
  - should distinctively identify local regions.
  - should allow
- the correspondence to overcomplete vocabulary of local features
  - need to come up with a robust, repeatable local image descriptor. You'll need to assign contrast strengths to the various features that map to this image piece. You'll want small changes in image appearance to yield small changes in the representation. Like, different sift features, all of which map to the same location,
  - How use the overcomplete set of sift feature matches at this position to explain all the details of this local image information stuff?

References

[1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH*, pages 187–194, 1999.

[2] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Pattern Analysis and Machine Intelligence*, 33(5), May 2011.

[3] S. M. Seitz and C. R. Dyer. View morphing. In *Proc. SIGGRAPH*, pages 21–30, 1996.

[4] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12:1247 – 1283, 2000.

[5] P. Weinzaepfel, H. Jegou, and P. Perez. Reconstructing an image from its local descriptors. In *Proc. IEEE Computer Vision and Pattern Recognition*, 2011.