Where computer vision needs help from computer science *

William T Freeman†

January, 2011

ACM-SIAM Symposium on Discrete Algorithms (SODA), January, 2011, invited talk.

Abstract

This paper describes areas and problems where computer vision can use help from the discrete algorithms community.

1 Introduction

Computer vision is a good target for the discrete algorithms of computer science. While we are far from being able to interpret images reliably using a computer, it is clear that there will be many benefits when we do reach that capability. Several aspects of the problem make it particularly appropriate for computer science research: we have large datasets of high-dimensional data, so efficient processing is crucial for success. The data are noisy, and we search and analyze images over Internet scales. Advanced algorithms research can help the field significantly.

The goal of this paper is to bridge fields. I want to describe the problems in a way that an outsider to computer vision can understand, and in particular, in a way that reveals the underlying discrete algorithm problems. I want to entice outside experts to help solve computer vision problems, and so I'll also give some tips at the end on how best to jump in.

This manuscript was partially crowd-sourced. At recent computer vision conferences, I've asked my colleagues where they felt we needed help from computer science and machine learning. Many of the points I present here are from those conversations, which I cite as "personal communication" with the appropriate researchers. Of course, any awkward descriptions or mistakes are mine.

*This research is partially funded by NGA NEGI-1582-04-0004, ONR-MURI Grant N00014-06-1-0734, Shell Research, and by gifts from Adobe, Google, and Microsoft.

†Computer Science and Artificial Intelligence Lab, 32 Vassar St., MIT, Cambridge, MA, USA

2 Computer vision today

To better explain where we need help, let me first mention what computer vision can do well. Under good conditions, computer programs can find and detect frontal-view faces almost as well as people can–most new cameras use that capability to control exposure and focus settings. In the controlled conditions of a factory, computers routinely detect defects in manufactured parts and labels. For example, most manufactured diapers are visually inspected by computer [34]. Computers read license plates and digitized documents, monitor traffic, and track traffic lanes from cars in highways.

But if you look more closely, even those successes reveal where much more progress is needed. Face recognition rates drop significantly for non-frontal faces, or under illumination change. Stereo algorithms can fail dramatically in the real world, failing to reconstruct the depth of large regions of the image, because of ambiguous correspondences or a lack of texture. It's thought that humans recognize thousands of object categories [4] but computers can only recognize a few categories reliably.

The curse of computer vision is variability, for which there are many sources. Variations in lighting conditions, viewpoint, and occlusion relationships change the observed image immensely. In addition, many different versions of the same thing simply look different: chairs take on many different forms and humans are always able to recognize them as such, but computers have more difficulty. Likewise, a material might look very different under different lighting or viewing conditions, yet people are very good at ignoring those variations and reliably perceiving the underlying material properties.

To illustrate current techniques, let me describe the current approach to object categorization. This will show how we convert the problem of computer vision into one where computer science tools can be used.

The biggest recent advance in computer vision has been the development of modern image features. What makes a good image feature? If we take a photograph under two different lighting or viewpoint conditions, an ideal image feature would return the same numerical description of the same image region under

those different conditions. Likewise, if we substituted one instance of an object class for another, for example one chair for another chair, we would want the feature descriptors to remain the same, so that we could use those descriptors to recognize object categories.

In the late 90's, researchers developed feature detectors which approximated those characteristics. The most successful of those are SIFT features (scale invariant feature transform), developed by David Lowe [31]. Through a combination of good design, good performance, and software made available online, these features have been phenomenally successful, and have been one of the true innovations in computer vision over the past decade. The 128 dimensional features are concatenations of histograms of local image orientations. Using orientation, rather than intensity, provides some insensitivity to lighting conditions [21]. The use of localized image histograms gives a rich description of the local image, while at the same time allows for slop in the precise configuration of image details. This works well to match common image regions seen from different vantage points, or even to identify examples of objects in the same category. These features may be measured at a set of locations that are characteristic in some way, or they may be measured over a dense grid of locations. SIFT features are used all over computer vision. They can be used for aligning images into a panorama, for finding instances of a set of specific objects, or as part of a system to recognize object categories. Often, the features are vector quantized into a vocabulary of several thousand "visual words".

At this point, the computer vision problems of image matching and object categorization begin to look like discrete algorithms problems. For image matching, we have a set of features f_i^a in image a, indexed by i, that we seek to match to the features f_j^b of image b. The spatial correspondences given by the matching features may determine a low-dimensional mapping between the images, called a homography, and that fact can be used to find reliable correspondences even with significant noise in the feature matches. Hypothesize-and-test algorithms such as RANSAC [17] are often used.

For the object class recognition task, we may have a training set of examples of many object categories, c. Each instance k of an object category generates some set of visual words, ${w^c}_k$. In general, the set of words w^c will vary from example to example within an object class, and because of differences in viewpoint, or lighting conditions, or because of occlusions. From the set of observed visual words, and comparison with the large, labeled training set, we want to efficiently infer which objects are present in the image, and where they are.

Relatively current methods for addressing these problems lead to performance such as shown in Fig. 1. The plots show object categorization performance as a function of training set size, on an easy dataset (Caltech 101) and a harder one (Caltech 256) [22]. On the harder dataset, of moderate complexity, the algorithm only shows performance of around 30%.

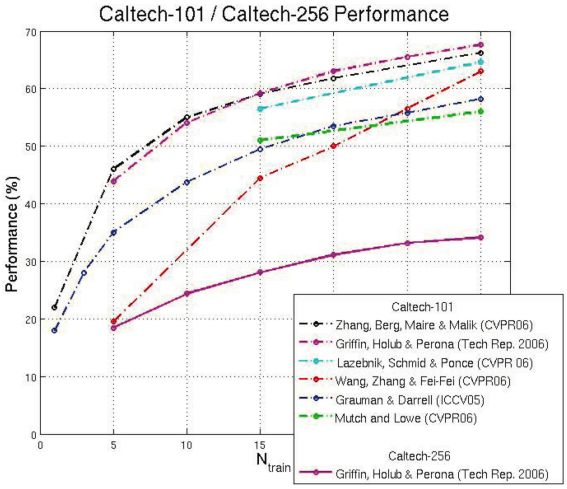

Figure 1: Object categorization performance as a function of training examples, for two different datasets [22].

The field makes progress over time, and current results are somewhat better [14], but we have a long way to go. Caltech 101 is by now considered overfit and results are no longer of interest. The PASCAL Challenge [13] is the current testbed for object class recognition research, but the best performances have been relatively flat over the past few years.

3 Where we need help

3.1 Approximate nearest neighbor search in high dimensions The object recognition task described above, and many other vision tasks, can be posed as one of nearest neighbor search in high dimensions: find the labeled training set examples that most closely match features seen in the test image. Indeed, the most common response from my computer vision colleagues to the question of how computer science can help computer vision was: find fast, approximate nearest neighbor algorithms that work well with high-dimensional data [43, 16, 28, 30, 25, 47, 36]. The items to be matched can be features, image patches, or entire images, but they are almost always high-dimensional. Current approximate methods, such as [1, 3, 35], are used, but improvements in speed and accuracy for high-dimensional data would give direct improvements to many computer vision algorithms.

Often the approximate nearest neighbor problems

we want solved have some additional structure. One may accumulate visual words over some region–we may have N feature words to match corresponding feature word collections from M candidate object classes in a large dataset of labeled examples. The feature word collections will have different sizes, and the matches will be noisy. How can we quickly identify the most probable object categories? How can we handle the feature variations between examples in a principled way? The image patches one may use are not random vectors of data, they are cropped images, so there is also structure within the vectors, and relationships between the elements. That is taken advantage of in some algorithms, such as patch match [3], but much more can be done.

Beyond this, we want to scale-up several aspects of the problem. For multi-class categorization, we want to successfully recognize several thousand, or even tens of thousands of object categories [4, 36]. The categories often reside in taxonomies, and we would like to take advantage of that structure. The unlabeled training set can be huge, of Internet scale. The labeled set, while large, will usually be much smaller. We need to generalize from the few labeled training examples, as discussed in [46]. We also want to scale-up current online methods for approximate nearest neighbor, and develop massive online quadratic programs for Support Vector Machines. [18, 27, 47] Many vision problems lead to integer programs, linear programs, quadratic programs, and semi-definite programs for large amounts of high-dimensional data. The standard solvers don't work and we need special purpose solvers that exploit the sparsity or structure of the problem [47].

3.2 Low-level vision While we sometimes process images to obtain a set of global labels–the name of some objects in the image–we can also process images to obtain labels, or a modified image, at every pixel. This can be thought of as "low-level vision" and typically uses a different set of tools. Prominent among those tools are Markov random fields. They are widely used, but still represent another area where computer vision can use help.

3.2.1 Markov random fields A Markov random field (MRF) is often used to describe images, since it provides a simple, modular desription, but can model a rich set of effects. For image applications, we typically use a grid structure. Let x_i be the unknown state at node i, and y_i be the observation there. The joint probability over the MRF is

$$P(\vec{x}, \vec{y}) = \prod_i \Phi(x_i, y_i) \prod_{i,j} \Psi(x_i, x_j), \tag{3.1}$$

where the second product is over neighboring nodes, i and j. $\Phi(x_i, y_i)$ can be thought of as a local evidence term. When the compatibility functions, described here as pairwise functions $\Psi(x_i, x_j)$ also depend on the observations, y, this becomes a Conditional Random Field (CRF). We often seek the set of states x_i at each node i that maximizes the joint probability $P(\vec{x}, \vec{y})$ for some given set of observations, $\vec{y}$, or, equivalently, that minimizes some function of the discrete variables, $\vec{x}$. Much progress has been made at this [5, 50, 48, 26, 45, 15], but what we can do is still limited by techniques available to solve this problem; advances would be immediately used in the community. We'd like efficient algorithms for minimizing non-sub-modular functions, and which give us bounds on the quality of the solution [47, 25]. There is real benefit to handling higher-order cliques in the MRF's and CRF's [26], modeling more than pairwise interactions, and we need better ways to do that.

Because topological constraints are often relevant in images, we often need to perform discrete optimization of Eq. (3.1) under such constraints. For example, we may want to specify that all states taking a particular label within some neighborhood should be connected, or that a user-specified bounding box should somewhere touch a member of some label set [29, 26]. We lack optimal or efficient ways to do that, or even to give bounds on performance [25]. In our graphical models, we often work with one of two kinds of constraints: structure constraints, such as planarity or treewidth, and language constraints, such as submodularity or convexity. It would be useful to be able to combine these two [25].

3.2.2 Statistical models for images Markov random fields are a statistical model of images, and has broad application in image enhancement and interpretation. While they have found much use, the field is struggling to develop even more useful image models. Other models have been developed [38, 37, 49], but evaluated by their utility for image synthesis, the best models are non-parametric texture synthesis algorithms such as [12, 11]. This raises the questions, can we do better than simply sampling from existing images to create a new image? Can we structure such non-parametric approaches to gain control over the image synthesis? Controllable models that created valid image samples would have broad application.

While feature detectors have contributed to progress in the field, there is probably much more that can be done, and the problem should be amenable to learning-based approaches. What is a natural encoding of images? We know that we'll need translation

and rotation invariance, so there's no need to discover that from data; there can be some supervision. Stepping back, an artist can indicate to us the trunk of an elephant with just a few strokes, but none of our mathematical features reach that level of simple efficiency [32, 36]. Issues of shape lead to more questions [36]: What is the right description for shape? How do we detect properties such as parallelism quickly? How do we detect other symmetries (e.g. center symmetry, axial symmetry etc).?

3.3 Miscellaneous Topics

3.3.1 Blind Vision Avidan and Butman introduced a vision-based cryptography problem, "Blind Vision" [2], allowing cooperation on a computer vision task between parties who don't want to share data or algorithms. For example, Alice may have a collection of sensitive surveillance images over which she would like to detect faces. Bob may have a proprietary face detection algorithm that he wants to let Alice use, for a fee. Blind vision applies secure, multi-party cryptographic techniques to vision algorithms so that Bob learns nothing about Alices surveillance images (not even the output of his own algorithm), and Alice learns nothing about Bob's face detection algorithm. Issues of commerce and privacy can be addressed through this line of research.

3.3.2 Compressed sensing While there is much excitement for the potential of compressed sensing [8, 9], the current sparsity assumptions are unrealistic for natural images [6, 30]. Is there a relaxed set of sparsity assumptions, that images meet, which would be useful for compressed sensing? Is there a useful application of compressed sensing in the domain of natural images?

3.3.3 Continuous to discrete From an engineering perspective, there are several concerns to address in using graphical models. For efficiency, multi-scale approximations to energy functions are often used, but this is done in an ad hoc way. What is the proper way to approximate a fine-scale energy function by a coarse-scale one [25]?

More ad hoc decisions are made in discrete-state representation of variables over a continuous domain [20]. For example, for stereo, a discrete-state graphical model is typically used to infer the depth at each position. Evidence for each depth state is gathered locally, then propagated across space using belief propagation or MRF energy minimization. How do the inferred states relate to those one would solve for using a continuous representation, or using a different number of discrete states? With linear signal processing, the relationship of discrete-domain to continuous-domain processing is well-understood: under assumptions about the spectral content of the signal, we know how to convert from the discrete to continuous domains to be guaranteed the same result in each domain. We would like to know how sharp or broad we should make the likelihood functions over continuous variables, such as depth. We should know how many discrete states to use, and how our discrete-state solution relates a continuous state solution.

Another representational issue relates to noisy evidence versus the lack of evidence. Presently, we usually treat those in the same way, but we might want to distinguish between there being a 10% similarity of something to a dog, versus a 10% probability that it's a dog, ie, to have a different description for weak relationship between things, and uncertainty about the relationship between things [10, 23]. It would be nice to allow for multiple different relationships between nodes–to have graphs with multi-colored edges [10].

3.4 Algorithms for learning and inference This is a broad category, but over the past 10 years, algorithms for inference or classification have swept through the field and had enormous influence. Support vector machines, boosting, belief propagation, and graph cuts have each led to much progress and creativity within the field. We're ready for the next one [47].

3.4.1 Handling large, noisy datasets Some people rank the relative importance of vision system components in this order: (1) datasets; (2) features; (3) algorithms [41].

Regarding datasets, we have three sources of data

1. hand-labeled data. This is slow and expensive to generate, and can be inconsistent.
2. unlabeled data. It is usually simple to acquire as much image or video data as might be desired.
3. synthetic data, fully labeled. Computer graphic images can approach photorealism, but it is difficult to fully model all the details and complexity of the real world.

We could use help with how to best combine those three sources of image information. Can we learn, from the labeled and unlabeled photographs, how to translate the rich, precise labeling of some computer graphic world to apply to photographic test data from the real world [39, 20]?

We also need progress handling our large, imperfect datasets. We always assume that our training and test distributions are the same, and they rarely are. Under

what circumstances can you break the assumption that those two distributions are the same [51]? Independent, identical distributions (IID) are generally assumed, but again is rarely satisfied for images and video. What is the effect on algorithms when the IID assumption doesn't hold [40]? The huge datasets we use, as well as online tracking problems, lead to a growing interest in online learning [47].

4 How to get involved

The "action" in computer vision happens at three IEEE-sponsored conferences: Computer Vision and Pattern Recognition (CVPR), the International Conference on Computer Vision (ICCV), and the European Conference on Computer Vision (ECCV). In addition, the series Foundations and Trends in Computer Vision [7] publishes a great series of long survey papers about particular topics in computer vision. It's a great way to get up-to-speed on a particular topic. I recommend these textbooks and journals: [19, 44, 24, 42]. The book that brought me into the field, Vision, by David Marr, has recently been re-issued by the MIT Press [33]. It's still a marvelous book and will get you excited about the field of vision.

Publishing at the top venues is difficult; they typically have 20-25% acceptance rates. By necessity, reviewers of submitted manuscripts are looking for reasons to reject, and they can't be as nurturing of some great, but unconventionally presented, idea as we would want them to be. To overcome this barrier, you'll need to collaborate with someone who is already in the field, or at least get their help in framing or writing your manuscript. You'll want to be reading and citing the relevant related work. The best way to get involved is to visit the labs of your computer vision colleagues. Our doors are open.

5 Acknowledgments

Thanks to the many researchers, named in the bibliography, who contributed their thoughts, and to those who commented on earlier versions of this manuscript or talk: Shai Avidan, Pushmeet Kohli, Jitendra Malik, Pietro Perona, Phil Torr, and Yair Weiss.

References

[1] A. Andoni, M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In T. Darrell, P. Indyk, and G. Shakhnarovich, editors, *Nearest Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, 2006.

[2] S. Avidan and M. Butman. Blind vision. In *in Proceedings of the 9th European Conference on Computer Vision*, pages 1–13. Springer, 2006.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *Proceedings of ACM SIGGRAPH*, Aug. 2009.

[4] I. Biederman. Recognition by components - a theory of human image understanding. *Psychological Review*, 94(2), 1987.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11), 2001.

[6] H. S. Chang, Y. Weiss, and W. T. Freeman. Informative sensing of natural images. In *in Proceedings of IEEE Intl. Conf. on Image Processing*, 2009.

[7] B. Curless, L. V. Gool, and R. Szeliski. Foundations and trends in computer graphics and vision, 2010.

[8] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4), 2006.

[9] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 2008.

[10] A. Efros, 2010. Personal communication.

[11] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *ACM SIGGRAPH*, 2001. In *Computer Graphics* Proceedings, Annual Conference Series.

[12] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Intl. Conf. on Comp. Vision*, 1999.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[14] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Pattern Analysis and Machine Intelligence*, 32(9), 2010.

[15] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *Intl. J. Comp. Vis.*, 70(1):41–54, 2006.

[16] R. Fergus, 2010. Personal communication.

[17] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[18] D. Fleet, 2010. Personal communication.

[19] D. A. Forsyth and J. Ponce. *Computer Vision: a modern approach*. Prentice Hall, 2003.

[20] W. T. Freeman, 2010.

[21] W. T. Freeman, D. Anderson, P. Beardsley, C. Dodge, H. Kage, K. Kyuma, Y. Miyake, M. Roth, K. Tanaka, C. Weissman, and W. Yerazunis. Computer vision for interactive computer graphics. *IEEE Computer Graphics and Applications*, pages 42–53, May/June

1998.
[22] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.
[23] D. Hoiem, 2010. Personal communication.
[24] IEEE.
[25] P. Kohli, 2010. Personal communication.
[26] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. In *Proc. IEEE Computer Vision and Pattern Recognition*, 2008.
[27] K. Kutalakos, 2010. Personal communication.
[28] S. Lazebnik, 2010. Personal communication.
[29] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *International Conference on Computer Vision*, 2009.
[30] A. Levin, 2010. Personal communication.
[31] D. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. Comp. Vis.*, 60(2):91–110, 2004.
[32] D. Lowe, 2009. Personal communication.
[33] D. C. Marr. *Vision*. MIT Press, 2010.
[34] D. Michael, 2010. Personal communication.
[35] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Intl. Conf. on Computer Vision Theory and Applications (VISAPP'09)*, 2009.
[36] P. Perona, 2010. Personal communication.
[37] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Intl. J. Comp. Vis.*, 40(1):49–71, 2000.
[38] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *Proc. IEEE Computer Vision and Pattern Recognition*, 2005.
[39] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.
[40] A. Saffari, 2010. Personal communication.
[41] A. Shashua, 2010. Personal communication.
[42] Springer.
[43] R. Szeliski, 2010. Personal communication.
[44] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.
[45] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Pattern Analysis and Machine Intelligence*, 30(6):1068–1080, 2008.
[46] J. B. Tenenbaum, T. L. Griffiths, and C. Kemp. Theory-based bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences*, 10(7):309–318, 2006.
[47] P. Torr, 2010. Personal communication.
[48] M. Wainwright, T. Jaakkola, and A. Willsky. Exact MAP estimates by (hyper)tree agreement. In *Adv. in Neural Info. Proc. Systems*, 2003.
[49] Y. Weiss and W. T. Freeman. What makes a good model of natural images? In *Proc. IEEE Computer Vision and Pattern Recognition*, 2007.
[50] J. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *International Joint Conference on Artificial Intelligence (IJCAI 2001)*, 2001.
[51] A. Zisserman, 2010. Personal communication.