PubMed Central
HHMI
HOWARD HUGHES MEDICAL INSTITUTE
AUTHOR MANUSCRIPT
Accepted for publication in a peer-reviewed journal

Published as: *Curr Opin Neurobiol*. 2010 October ; 20(5): 653–666.

Machines that learn to segment images: a crucial technology for connectomics

Viren Jain, **H Sebastian Seung**, and **Srinivas C Turaga**
Howard Hughes Medical Institute and Brain & Cog. Sci. Dept., Massachusetts Institute of Technology, Cambridge, MA 02139, United States

Abstract

Connections between neurons can be found by checking whether synapses exist at points of contact, which in turn are determined by neural shapes. Finding these shapes is a special case of image segmentation, which is laborious for humans and would ideally be performed by computers. New metrics properly quantify the performance of a computer algorithm using its disagreement with 'true' segmentations of example images. New machine learning methods search for segmentation algorithms that minimize such metrics. These advances have reduced computer errors dramatically. It should now be faster for a human to correct the remaining errors than to segment an image manually. Further reductions in human effort are expected, and crucial for finding connectomes more complex than that of *Caenorhabditis elegans*.

Imaging technologies have influenced biology and neuro-science profoundly, starting from the cell theory and the neuron doctrine. Today's golden age of fluorescent probes has renewed the belief that innovations in microscopy lead to new discoveries. But much of the excitement over imaging overlooks an important technological gap: scientists not only need machines for making images, but also machines for *seeing* them.

With today's automated imaging systems, it is common to generate and archive torrents of data. For some experiments, the greatest barrier is no longer acquiring the images, but rather the labor required to analyze them. Ideally, computers would be made smart enough to analyze images with little or no human assistance. This is easier said than done — it involves fundamental problems that have eluded solution by researchers in artificial intelligence for half a century.

One of these problems is *image segmentation*, the partitioning of an image into sets of pixels (segments) corresponding to distinct objects. For example, a digital camera user might like to segment an image of a room into people, pieces of furniture, and other household objects. A radiologist may need the shapes and sizes of organs in an MRI or CT scan. A biologist may want to find the cells in a fluorescence image from a microscope. Engineers have tried to make computers perform all of these tasks, but computers still make many more errors than humans.

Recently there has been progress in answering two basic questions about image segmentation.

Corresponding author: Seung, H Sebastian (seung@mit.edu).

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

HHMI Author Manuscript
HHMI Author Manuscript
HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

1. Given two different segmentations of the same image, how can the amount of disagreement between them be quantified?
2. Given a space of segmentation algorithms, how can a computer be used to search for a good algorithm?

In the past few years, the first question has been addressed by the introduction of metrics that mathematically formalize our intuitive notions of ‘good’ segmentation. These metrics penalize topological disagreements between segmentations, and are less sensitive to small differences in boundary locations [3•,4••]. The new metrics are significant, because they can be applied to quantify the performance of a computer algorithm by measuring its disagreement with ‘true’ segmentations of a set of example images (generally provided by humans). Good metrics are absolutely essential for progress in research. Without them it is not even possible to tell whether progress is being made.

The second question has been answered by formulating the search as an optimization. Use a computer to search for an algorithm that minimizes disagreement with the true segmentations, as measured by the new metrics [5••,4••]. Such automated search is called *machine learning* from examples. It is distinct from the conventional approach, in which a human directly designs a good algorithm using intuition and understanding. Many still adhere to the conventional approach, which has produced a huge number of papers over decades of research.[a] But empirical results have shown that machine learning produces superior accuracy (see the Berkeley Segmentation Benchmark at http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/html/algorithms.html). The utility of machine learning has already become accepted for other computer vision tasks such as object recognition [6], and we expect that it will become standard for image segmentation as well.

The above innovations are quite general, but applications to just two image domains will be discussed in this review. Research on segmenting ‘natural’ images, or photographs of ordinary scenes, began in the late 1960s [7]. Research on segmenting serial electron microscopic (serial EM) images of neurons began in the 1970s [8], and has largely applied algorithms or ideas that were originally developed for natural images. But in the past few years, this niche area has given rise to innovations that have yet to be applied to natural images.

Note that natural images are two-dimensional (2d), while serial EM images are three-dimensional (3d). The ideas discussed in this review are applicable to both types of images, and more generally to arbitrary dimensionality. Serial EM produces a 3d image one slice at a time, generating a ‘stack’ of 2d images [9]. By segmenting serial EM images of neurons, one can find their shapes, including the trajectories of their axons and dendrites. The shapes of neurons are important because they determine whether neurons contact each other. By checking all contact points for synapses, it is possible to map all the connections between neurons, to find a *connectome* [10]. This process was carried out for the nematode *Caenorhabditis elegans* in the 1970s and 1980s [11]. Although the *C. elegans* connectome contains just 7000 connections between 300 neurons, it took over a decade to find. Most of the time was spent on image analysis, which was performed without the aid of computers.

Recent advances in serial EM [12–15] have revived interest in finding connectomes. These improved methods promise to produce images of larger volumes of brain tissue.[b] A cubic millimeter is estimated to require up to hundreds of thousands of person-years of human effort

[a] A Google Scholar query for the phrase ‘image segmentation’ yields over 200 000 references.

[b] The FIB-SEM method also produces high resolution images, but it is not yet clear whether it can be scaled up to larger volumes [16].

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

to segment manually [17]. From such numbers, it is obvious that the need for automated segmentation has become even more acute.

This review focuses only on image segmentation. We will not address the automation of synapse detection, because this important problem is little studied so far. Arguments for the importance of connectomes to neuroscience can be found elsewhere [18,19]. Finally, we do not address reconstruction of isolated neurons from the sparse images generated by light microscopy. In the limit of well-isolated neurons, this is not a problem of segmenting multiple objects, but rather of finding the best description of a single object as a tree.

The segmentation problem

The following two definitions of the segmentation problem are equivalent.

Definition 1

Segmentation as partitioning Partition the image into sets of pixels called segments, which correspond to distinct objects.

Definition 2

Segmentation as an equivalence relation Decide whether each pair of pixels belongs to the same object or different objects.

Definition 1 is more intuitive to most people, while Definition 2 is useful for some of the formalism described below. The definitions are equivalent because of the mathematical fact that any partitioning corresponds to an equivalence relation.

It is common to display the result of segmenting an image by a *region coloring*, which assigns colors to the pixels of an image, such that different colors correspond to different objects. Example colorings are shown in Figure 1. A coloring may reserve a special color for pixels which do not belong to any object. These pixels belong to boundaries between objects, or the background. It is trivial to turn a coloring into a partitioning: two pixels belong to the same segment if and only if they have the same color. A coloring is a nonunique representation of a segmentation, since any permutation of colors leads to the same partitioning.[c]

Boundary detection

As their first stage, many segmentation algorithms perform the computation of

> *Boundary detection*. Decide whether each pixel belongs to a boundary between objects.

The result of this computation is a *boundary labeling*, a black-and-white image in which white pixels correspond to boundaries, and black pixels correspond to interiors of objects (see Figure 1).

A second stage transforms the boundary labeling into a segmentation (as in Definition 2) by using *connectedness* as an equivalence relation between pixels. Two interior pixels are said to be *connected* if there exists a path between them that traverses only interior pixels in the

[c] A coloring is also a nonlocal representation, in the sense that the coloring of different pixels cannot be done independently. Imagine a thought experiment in which every pixel in an image is assigned to a different person. Even if all people know the correct segmentation, there is no way for them to indicate it through coloring, unless they communicate with each other. This is not the case for boundary detection or affinity graph labeling.

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

boundary labeling. Connected sets of interior pixels (connected components) correspond to segments of a partitioning or coloring.

Separating the segmentation computation into two stages is natural, because the two stages involve nonlocal processing to different degrees. The second stage of partitioning is inevitably nonlocal because it involves finding out whether pairs of pixels are connected, and these pixels may be distant from each other. And even if the two pixels are nearby, the path connecting them might travel arbitrarily far away.

As shown in Figure 1, the first stage of boundary detection would ideally be nonlocal also, because there are difficult locations in the image where boundaries cannot be accurately detected without contextual information from distant pixels. But for most locations, nearby pixels are sufficient for making the correct decision. Therefore, many boundary detection algorithms consider only local information. This constraint limits accuracy, but it also improves speed — a practical compromise. Local boundary detectors generally look for abrupt changes in various properties such as intensity, color, and texture [20,21,8]. Such algorithms are said to be *gradient-based*, because the abrupt changes are found by thresholding some kind of spatial derivative.

Because local boundary detectors are quite inaccurate (Figure 2), their output is often fed to a subsequent stage of computation that is supposed to use contextual information to correct the errors. The Canny edge detector was a simple version of this idea [22]. More sophisticated methods included relaxation labeling [23], nonlinear diffusion [24], Markov random fields [25,26] which can sometimes be optimized efficiently using graph cuts [27], active contours [28], and level sets [29,30].

These methods all involve interesting mathematics, and their proponents like to focus on the differences between them. We prefer to regard the methods as more similar than different. All define a dynamics (or an optimization) of auxiliary variables associated with the pixels. Each variable is updated depending on a linear combination of variables from neighboring pixels, as well as some kind of nonlinear operation. Iteration of the dynamics propagates information over long distances.

Nonlinear diffusion [31], Markov random fields optimized by graph cuts [32,33•], level sets [34,35•,36,37], and active contours [38–41] have also been applied to EM images of neurons, mostly in the last decade.

In practice, the above algorithms generate analog values rather than binary labels. These values are sometimes interpretable as the *probability* that a pixel belongs to a boundary. The analog boundary labeling, or boundary map, can be thresholded to produce a binary labeling, which is then used to find connected components.

In EM images, even a small rate of missed boundary pixels in the boundary map can result in an undersegmentation by the connected components procedure. In practice, this is prevented by using a low threshold for boundary detection resulting in a large false positive rate, but a low false negative rate, and thus fewer mergers [42••].

The regular watershed algorithm is an alternative approach to creating a segmentation from a boundary map [43]. This approach is distinct from connected components, but is closely related. The watershed algorithm tends to oversegment the image, producing many more segments than objects. This is because there is a watershed domain for each local minimum of the boundary map, and local minima are typically very numerous. Therefore, watershed is generally augmented by schemes for damping local minima or merging watershed domains to form larger segments.

HHMI Author Manuscript
HHMI Author Manuscript
HHMI Author Manuscript

Affinity graph labeling

Boundary detection is not the only possible first stage for a segmentation algorithm. An alternative is to label the edges of an *affinity graph*, which consists of nodes corresponding to image pixels.

> *Affinity graph labeling*. Label each affinity graph edge to indicate whether its pixels belong to same or different objects.

Each edge label is called an *affinity*. As with a boundary labeling, a second stage of computation is required to transform the affinity graph into a segmentation. This second stage has two goals.

The first goal is the resolution of inconsistencies. Suppose that an affinity graph is fully connected, containing all possible edges between nodes. Then labeling its edges would seem to specify an equivalence relation between pixels, as in Definition 2 of segmentation. But the edge labels may violate the property of transitivity, and hence be inconsistent with an equivalence relation. These inconsistencies must be resolved to produce a segmentation.

The second goal is to supply missing information. If the affinity graph is only partially connected, then it only partially specifies an equivalence relation. Therefore the second stage of computation must decide about the missing edges of the graph, as well as resolve inconsistencies.

Both of these goals can be accomplished by defining connectedness in the graph to be an equivalence relation. Two nodes are said to be connected in the graph if there exists a path between them traversing only edges with affinity equal to one. The affinity graph is partitioned into connected sets of nodes by the second stage of computation.

Various types of connectivity have been used in affinity graphs. One type of partial connectivity contains edges only between adjacent or nearest neighbor (NN) pixels. Such a NN graph has been used for EM images [44••] as well as natural images [45]. A NN affinity graph is quite similar to a boundary labeling.[d] The only difference is that boundaries are located at the midpoints between pairs of adjacent voxels, rather than at the voxels themselves. Adjacent voxels can belong to different objects in a NN affinity graph, but not in a boundary labeling.[e]

This can be advantageous for representing segmentations of EM images with limited spatial resolution. Where neurites become very thin, there may not be enough voxels to represent both the interiors of the neurites as well as the boundaries between them.[f] This problem is solved by representing boundaries using edges between voxels (Figure 3a). Also, if the 3d image is composed of aligned 2d images of physical slices, errors in alignment can cause voxels of two different cells to end up adjacent to each other across two slices, with no extracellular space between them (Figure 3b).[g]

An affinity graph can also contain edges between pairs of voxels that are not nearest neighbors. Then it is less like a boundary labeling. An affinity graph can represent a segment that consists

[d] But note that an affinity graph has the opposite sign convention as a boundary labeling, which can be confusing.

[e] In a boundary labeling, boundaries are represented by voxels that do not belong to any cell, but rather to extracellular or 'outside' space. A NN affinity graph does not need to assign voxels to boundaries, since it uses edges to represent boundaries. However, it has the option of assigning voxels to extracellular space, in which case they end up being disconnected from each other and from all cells [44••]. This means that a NN affinity graph is more powerful than a boundary labeling, in the sense that it can represent more partitionings of the image. Of course, this power is achieved by including more information: there are more edges in a NN affinity graph than voxels in a boundary labeling.

[f] Here the true interpretation of the image may be ambiguous based on local information, but can become unambiguous when more context is included.

[g] Another way of dealing with this problem is to compute a super-sampled output image that has higher resolution than the input image [46•].

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

of a set of voxels that is disconnected in the image, as long as the set is connected in the graph. This is useful for EM images with limited spatial resolution, as thin neurites can ‘break’ when they become less than one voxel in diameter. Although such neurites are disconnected in the image, they can be connected by long-range edges in the affinity graph. Similarly, if the 3d image is composed of aligned 2d images of physical slices, errors in alignment can cause a thin neurite to end up disconnected in the image (Figure 3c).[h]

All these advantages of the affinity graph can be summarized in a single bottom line: connectedness is based on the definition of adjacency, and this definition is more flexible in an affinity graph than in a boundary labeling.

As with boundary detection, achieving the highest accuracy at labeling an edge of an affinity graph would require contextual information from distant pixels in the image. But in practice, local algorithms are often used for the sake of speed. For nearest neighbor edges, the definition of local is basically the same as it was for boundary detection. For labeling long-range edges, an algorithm is said to be local if its output depends only on the two image patches surrounding the two pixels connected by the edge. The affinity is generally computed using some measure of similarity of the two image patches, based on properties such as intensity, color, and texture [45].

As with boundary labelings, it is common for algorithms to generate analog values rather than binary labels for the affinity graph edges. These values are sometimes interpretable as the *probability* that an edge connects two pixels that belong to the same object. These analog values can be thresholded to produce a binary edge labeling, which is then used to find connected components.

In addition to connected components, many other algorithms have been proposed for partitioning an affinity graph. These are supposed to correct the errors and inconsistencies produced by local computation of the affinity graph labeling. Spectral methods have been applied to produce a hierarchical partition of the affinity graph [47]. An analog of the watershed algorithm can be defined for graphs through the minimum spanning tree [48,49].

Manual creation of segmentation datasets

Two broad classes of segmentation algorithms were defined above, those that involve boundary detection and affinity graph labeling. Rather than describing more classes of algorithms, we move now to a different subject, that of evaluating performance. Surprisingly, this issue was not confronted seriously until the 2000s. Previously, researchers had evaluated algorithms subjectively, by inspecting performance on a few images. Without objective and quantitative means of evaluation, it was difficult to tell which algorithms were better.

Usually it is easy to evaluate performance in computer science. Many computational tasks, like multiplying numbers or inverting matrices, have simple and precise mathematical specifications, so that it is straightforward to measure speed and accuracy. But no explicit specification exists for image segmentation and other tasks in computer vision. There is no way to evaluate performance without resorting to empirical means.

This was first done in a systematic way by the introduction of the Berkeley Segmentation Dataset [1•]. Its creators collected natural images, and employed humans to segment them. Both images and segmentations were made publicly available. Researchers could quantify and

[h] In natural images, partial occlusion can split a background object into regions that are disconnected in the image. Such an object can still be connected in an affinity graph with long-range edges.

HHMI Author Manuscript HHMI Author Manuscript HHMI Author Manuscript

compare the performance of their computer algorithms by measuring disagreement with the human segmentations using a common dataset. More recently, researchers have created similar datasets that contain EM images along with segmentations [46•,5••,44••,4••,50•]. So far one of them is publicly available [50•].

To manually segment a 2d image, a human can use standard computer software for drawing or painting. In *boundary tracing*, the human draws contours at the boundaries of objects. In *region coloring*, the human paints the interiors of objects with different colors. These methods can be extended to 3d images by allowing the user to annotate 2d slices.

The software packages RECONSTRUCT [51], Tra-kEM2 [52], and KLEE (Moritz Helmstaedter, personal communication) are capable of boundary tracing of EM images. As illustrated in Figure 1, ITK-SNAP [53] can be used for region coloring of biomedical images.

For segmenting neurons, there is an alternative that is faster and captures most of the shape information. A human can draw a line along the axis of a neurite, with the capability of adding branch points. This functionality is implemented by the software packages TrakEM2, Elegance (S Emmons, unpublished data) and KNOSSOS (M Helmstaedter, unpublished data). Such *skeleton tracing* is more than ten times faster than full segmentation [17]. SSECRETT [37] and Piet (J Lichtman, unpublished data) allow thedroppingof 'bread crumbs,'in orderto summarize neurons as sets of points. This is a bit less information than skeletons, which include the lines connecting the points.

Neuro3D [34] and NeuroTrace [36,37] are software packages for semi-automated tracing of neurites that combine active contours or level sets with a graphical user interface. For each 2d image slice, the computer suggests a contour for the boundary of the neurite. The human user either accepts this contour or corrects it. NeuroTrace utilizes parallel computation with GPUs to speed up the level set computation.

Metrics of segmentation performance

The introduction of common segmentation datasets is essential for allowing researchers to properly quantify and compare the performance of their computer algorithms. But datasets alone are not enough. It turns out that defining a proper metric for measuring disagreement between segmentations is a nontrivial problem. Only recently have good solutions been proposed.

In general, a metric can be used to compare any pair of segmentations. Most commonly, one of the segmentations comes from a computer, and the other from a human. If the human segmentation is regarded as the 'truth,' then the metric measures the error of the computer. Therefore we will often use the term 'error' interchangeably with 'metric.' Metrics can also be used to compare two human segmentations. Consistency of human segmentations is an important indication of whether it makes sense to regard them as the 'truth' [46•,1•,54••].

Figure 4 illustrates the difficulty of defining a good metric. A human segmentation of an EM image is compared with the segmentations produced by two hypothetical computers. Which computer is better? A naive method of evaluation is to count the number of pixels on which the computer boundary labelings disagree with the human boundary labeling. By this metric, the *pixel error*, the two computers are equally good. But this evaluation is inconsistent with our intuitive notion of a good segmentation. In computer segmentation *A*, the red object in the human segmentation is missing, the yellow object in the human segmentation is split into two objects, and the blue and green objects in the human segmentation are merged into one. In other words, computer segmentation *A* contains three errors: a deletion, a split, and a merger. (There is also a small hole, which would not lead to an error in finding connectomes.) A split

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

is caused by the incorrect detection of a spurious boundary, while a merger is caused by an incorrect gap in a boundary. Computer segmentation *B* contains the same objects as the human segmentation. The shapes are slightly different, but there are no genuine disagreements at all. In short, Computer *B* is intuitively superior to Computer *A*, but the pixel error does not reflect that.

Figure 4 suggests that the ideal metric should

- tolerate minor differences in boundary location,
- and strongly penalize topological disagreements like splits and mergers.

For image segmentation in general, we have appealed to intuition to motivate these properties. But their importance is even more clear in the particular application of connectomics. As mentioned earlier, split and merger errors have serious consequences, causing the many connections on a stretch of neurite to be erroneous. In contrast, an error in boundary location can cause an error in detecting a synapse at that particular location. Furthermore, the error in boundary localization must be so large that it prevents the synapse from being assigned to the correct neuron.

The pixel error has neither of the two ideal properties above. The Berkeley Segmentation Dataset was introduced with a set of metrics that have the first property [1•]. But the Berkeley metrics are unsatisfactory because they may not penalize topological errors (such as small gaps in a boundary) that could lead to large differences in segmentations [55,4••]. Indeed, when applied to the segmentations in Figure 4, the Berkeley metrics assign *lower* error to the intuitively worse interpretation of Computer *A*.

A new metric called the *warping error* [4••] possesses both of the desired properties listed above. To compute the metric, the human boundary labeling is warped to match the computer boundary labeling. This is done by flipping pixels of the human boundary labeling that do not change its topology, and also reduce disagreement with the computer. The topological constraints are enforced by using the idea of a simple point, borrowed from the field of digital topology [56]. A geometric constraint is also imposed by not allowing boundaries to shift more than some distance cutoff. After warping is complete, the remaining pixel disagreements constitute the warping error. In Figure 4, the warping error of Computer *A* consists of just five pixels, all true topological disagreements with the human. Computer *B* has zero warping error, conforming to our intuitive notion that it is superior to Computer A.

The *Rand error* is a second metric with the two desired properties. It was originally proposed for data clustering [57], and only recently adopted as a metric for image segmentations [3•]. Let us say that two pixels are ‘connected’ in a segmentation when they belong to the same region. The Rand error is defined as the fraction of pixel pairs that are connected in one segmentation but not in the other. A split or a merger produces large Rand error, while a small shift in boundary location produces little Rand error. In Figure 4, Computer *B* is again superior to Computer *A* by the Rand error (data not shown), which matches our intuitive ranking.

Machine learning from examples

Earlier we listed a number of conventional algorithms for image segmentation. Such algorithms are found through a collective search conducted by a community of many human experts. Each researcher proposes new algorithms and compares them with old ones. The new metrics described above make it possible to perform the comparisons properly.

But quantitative metrics enable a different approach to research: use a computer to automatically search for new and better algorithms. This *machine learning* approach

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

- defines a space of algorithms that take images as input and produce segmentations as output,
- and instructs the computer to 'learn,' that is search the space for an algorithm that optimizes a metric of segmentation performance.

This could be explained to corporate executives as 'management-by-objective.' Instead of telling the computer how to segment images, we quantitatively define the objective of image segmentation. The computer figures out its own way to achieve that objective.

Some researchers have resisted the machine learning approach. They prefer to design algorithms directly, based on their intuitive understanding of image segmentation. But machine learning has yielded superior performance in the Berkeley Segmentation Benchmark [1•]. Perhaps this is because our intuitions about image segmentation are not particularly good. Although we are conscious of the *results* of visual computations by our brains, many of the processes leading to these results are unconscious and inaccessible to introspection. This is the reason that visual psychologists and neuroscientists have to struggle to understand how our brains perform visual tasks, though vision itself is effortless for us.

Research on machine learning of image segmentation initially used the naive metric of the pixel error. A classifier was trained to perform the task of boundary detection by measuring its pixel error relative to a dataset of true boundary labelings. Even with this primitive metric, machine learning delivered superior performance at segmenting natural images as compared to the classic Canny edge detector and a second-moment matrix based detector related to corner detection techniques [1•,58•]. Studies on EM images have also shown the superiority of machine learning as compared to Hessian-based ridge detection and anisotropic smoothing [42••,54••,59,33•,60•,61].

In management-by-objective, it is crucial to define the objective correctly, lest employees try to achieve the wrong goals. Similarly, it is crucial for machine learning to optimize a good performance metric. As we saw earlier, pixel error is not a good segmentation metric. The Rand and warping errors are metrics that formalize our intuitive notions of good segmentation. These metrics have stimulated the latest phase of machine learning research on image segmentation. Maximin Affinity Learning of Image Segmentation (MALIS) trains a boundary detector by minimizing its Rand error [5••]. Boundary Learning by Optimization with Topological Constraints (BLOTC) trains a boundary detector by minimizing warping error [4••]. Recent work suggests that these new machine learning methods improve accuracy substantially over machine learning based on pixel error.

Designing versus learning features

Above we have portrayed the machine learning approach as searching for an algorithm that transforms the input into the desired output.

$$\text{input} \xrightarrow{\text{learned}} \text{output}$$

It is common to break this transformation into two stages.

$$\text{input} \xrightarrow{\text{designed}} \text{feature vector} \xrightarrow{\text{learned}} \text{output}$$

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

The first stage is designed by hand, and computes a ‘feature vector,’ the components of which signify the presence or absence of various features in the input. Only the transformation of the feature vector into the desired output is learned.

The first stage embodies the designer’s understanding of the computation, while the second stage encapsulates the designer’s ignorance. If the designer’s understanding is fairly complete, a simple algorithm will suffice for the second stage. Machine learning is easier if the class of algorithms to be searched is simpler, for two reasons. First, searching for an algorithm that performs well on the training set generally takes less time (less computational complexity). Second, the result of the search tends to generalize better to novel inputs. This means that fewer examples need be collected for the training set, which can be a major reduction in human labor (less sample complexity). Intuitively, it makes sense that learning will be easier when it takes advantage of prior knowledge.

In areas such as speech recognition [62,63] or visual object recognition [64], researchers have devoted significant effort to designing good features. Research on segmenting natural images has also identified good features, such as gradients of intensity, color, and texture [1•]. Research on EM image segmentation has mostly relied on features that were originally introduced for natural images [42••,54••,59,33•,61], although recent work has introduced features specifically designed for EM image analysis [65,66•]. The transformation of the feature vector into the boundary map has been learned by a number of methods, including random forest [42••,33•], boosting [59], and multi-layer perceptrons [54••,61].

A less common approach is to learn the entire transformation from input to desired output, dispensing with a hand-designed first stage. This is sometimes called ‘end-to-end’ learning [67].[i] This approach is attractive when the researcher lacks sufficient understanding to design good features. Furthermore, it is not adversely affected if the researcher’s intuitions about the computational task are actually incomplete or erroneous. End-to-end learning has the disadvantage that it may require more examples and more computational time. However, training sets for EM images can contain millions of labeled voxels, which appears to be sufficient for producing superior generalization performance with end-to-end learning [4••].[j] Furthermore, computers are now fast enough to make such learning practical.

A convolutional network (CN) is a convenient and powerful class of algorithms for end-to-end learning of image segmentation [72•,46•,44••]. A CN is organized in layers, each of which performs a set of linear convolutions and pixel-wise nonlinear transformations.[k] A CN can implement gradient-based boundary detection by appropriate choice of the convolution filters. A CN can also perform approximate inference for Markov random field models [46•]. Therefore machine learning based on CNs is likely to perform at least as well as these relatively simple hand-designed algorithms. Since CNs may employ hundreds of filters that involve tens of thousands of free parameters [46•,5••,44••,4••], machine learning may also find a more complex algorithm that significantly outperforms the ones designed by hand. Indeed, this has turned out to be the case in some empirical studies [46•,4••]. While CNs can be time-consuming to train, this problem is somewhat alleviated by fast GPU implementations [73].[l]

[i] In a third approach the first stage is found by ‘‘unsupervised learning.’’ Features are learned from images by optimizing an objective function that does not depend on the desired output of the ultimate task to be learned [68–71]. The ‘‘supervised learning’’ discussed in this review is only applied to the second stage.

[j] Active learning is another approach to minimizing the use of training data. The human traces boundaries at image locations selected by the computer, rather than tracing every location unselectively.

[k] Note that most research on CNs has used them to implement image-to-label transformations [72]. These networks are more precisely called convolution-subsampling networks, as they employ subsampling to discard positional information.

[l] While CNs require long training time, once trained they are faster than many other boundary detectors when applied to novel images.

HHMI Author Manuscript HHMI Author Manuscript HHMI Author Manuscript

Designing features is analogous to micromanagement, while learning them is analogous to pure management-by-objective. In fact, there is a spectrum between fully learned and fully designed. Future research should be able to identify the best mixture of learning and design, which may vary depending on the particular application. It will be important to use good metrics like the Rand or warping error to decide between the relative merits of various approaches.

Harnessing human effort efficiently

Let us shift now from fundamental ideas in computer vision to their application in practical systems for connectomics. Suppose that we would like to segment a large dataset of EM images. Note that a single segmentation error can lead to a large number of erroneous connections in a connectome. For example, if an axon is connected with the wrong cell body, then all of its synapses will be erroneously assigned to the wrong neuron. Unfortunately, state-of-the-art segmentation algorithms still make many errors per neuron. In short, connectomics requires extremely accurate segmentation, and current algorithms are far from achieving this. Much more research will be required to fully automate image analysis.

In the near future, semiautomated segmentation will be the best strategy[m]:

1. Humans manually segment a subset of the data to create a training set.
2. Machine learning is applied to find an image segmentation algorithm.
3. The algorithm is applied to the rest of the image data to produce a candidate segmentation, which contains errors.
4. Humans correct the errors, editing the candidate segments through split and merge operations.

Making this semiautomated pipeline efficient means minimizing human effort, which is mainly consumed by Steps 1 and 4. For the pipeline to be useful, it must consume less human effort than fully manual segmentation. The problem of reducing the human effort in Step 1 was discussed in the previous section on design versus learning of features.

For the final ‘editing’ of Step 4, special software is necessary for allowing humans to interact with the computer, and perform the splitting and merging operations. Such software is still in its infancy, so there are few quantitative results about human effort in the literature. One study has claimed greater than tenfold reduction in human effort compared to manual segmentation [54••,84]. Such comparisons are encouraging, but they should be regarded as preliminary. Even estimates of the speed of manual segmentation vary over a tenfold range [17], so further studies will be needed for a clearer picture.

Note that the Rand and warping errors used in Step 2 can be regarded as proxies for the human effort consumed by Step 4, which is expected to be roughly proportional to the number of split and merge errors to be corrected.

Step 4 poses another interesting challenge, which is to develop methods and software that allow multiple humans to cooperate by interacting with the computer to generate segmentations. This could enable higher accuracy by averaging out the ‘noise’ in the judgments of individual humans. ‘Crowdsourcing’ could also be used to recruit larger numbers of humans over the Internet to edit computer segmentations [74,75].

[m] A more limited use of the computer is to automatically generate full segmentations starting from manual skeleton tracings. ‘Putting flesh on the bones’ can require less human effort than generating full segmentations manually, and is important for identifying points of contact between neurons (M Helmstaedter, unpublished data).

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

Learning to split and merge

In the semiautomated pipeline described above, Step 4 is performed by humans. It would make sense to automate this step also — so that computers perform the merge and split operations. Most efforts along these lines have used a first stage of boundary detection to generate an oversegmentation. In other words, the computer is made to err on the side of splitting, producing only small fragments of objects. These fragments are sometimes called ‘superpixels’ [76] or ‘supervoxels’ [42••]. Then the goal is to merge fragments to form correct segments.

This approach is increasingly common for natural image segmentation [76–80]. For EM image segmentation, some researchers have found superpixels in each 2d slice of an image stack, and then designed decision criteria to merge superpixels across slices to generate 3d objects [81•, 41,54••,82,60•].

One can also apply machine learning to this problem, using human-generated merge and split operations as training data. Andres *et al.* performed boundary detection with a random forest classifier, and merged the resulting 3d supervoxels using a second random forest classifier [42••]. Such work is still in its infancy. Further research is needed on models for making split and merge decisions, and on applying machine learning to such models.

While an oversegmentation is a popular starting point, another possibility is to start from a candidate segmentation with a more even distribution of split and merge errors.[n] This strategy could be more efficient, whether the split and merge operations are performed by human or computer.

Outlook

New performance metrics, as well as machine learning methods based on these metrics, are transforming research on image segmentation. These innovations have largely been driven by the goal of segmenting serial EM images of neurons. One might ask why this niche application has played a disproportionately important role. One reason is that the shapes of neurons are highly complex, making accurate segmentation extremely difficult, and forcing researchers to try new ideas. A second reason is that a serial EM image actually possesses a ‘true’ segmentation, which neuroscientists really want to know. In contrast, the notion of a segmentation is not completely well-defined for natural images, as evidenced by the fact that human segmentations of images in the Berkeley dataset often disagree substantially. Researchers may not have a strong incentive to achieve very low error rates, since this may be fundamentally impossible anyway.

As described above, machine learning approaches have created new algorithms for segmenting EM images with outstanding accuracy relative to conventional algorithms. For the first time, semiautomated segmentation is becoming faster than purely manual segmentation (though more precise quantification of this claim is needed). This is an important milestone, demonstrating the utility of computerized image segmentation, but it is just the beginning. It is important to further reduce human effort consumed by semiautomated segmentation. How can computer accuracy be improved even further? And more fundamentally, why should we believe that further gains are possible at all?

As mentioned earlier, boundary detection usually starts with a local computation that is based on a limited field of view. Accuracy is improving due to machine learning, but performance will eventually saturate due to difficult locations that are inherently ambiguous. There exists

[n] Here also the candidate segments could be called ‘superpixels’ or ‘supervoxels,’ although in the original meaning this term was defined as an indivisible group of voxels [76].

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

no algorithm that can perform well at these locations based on a limited field of view, so machine learning cannot be expected to find one.

In principle, the solution to this problem is simple: increase the field of view. If the boundary detector is allowed to use more contextual information, some of the difficult locations will be disambiguated. Research on EM image segmentation is likely to progress by steps that increase the field of view. At each step, performance will increase as researchers succeed in exploiting the new contextual information. When performance saturates, the field of view will be increased again.

What are the challenges involved in increasing the field of view? First, the computation may become slower, due to the demands of processing more information. In 3d, doubling the length of the field of view increases the volume by almost an order of magnitude.

Second, a larger field of view will not simply provide more contextual information, but information of a different type. Utilizing it may require a computation that is quite different. Therefore, it is unrealistic to proceed by simply scaling up a single monolithic computation to an arbitrarily large field of view.[o] One possibility is to modify existing boundary detection techniques to perform multi-scale computation [83], by combining separate computations at multiple spatial resolutions.

If boundary detection is followed by another stage of automating splitting and merging of supervoxels, this could potentially provide an efficient and powerful means of dealing with added context. Each supervoxel should be represented by a descriptor more compact than its raw voxels. Ideally, future research will yield shape descriptors that allow fast and accurate decisions about split and merge operations.

References and recommended reading

Papers of particular interest, published within the period of review, have been highlighted as:

• of special interest

•• of outstanding interest

1•. Martin DR, Fowlkes CC, Malik J. Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans Pattern Anal Mach Intell 2004:530–549. Introduces the first boundary detection algorithm that incorporates machine learning, with application to natural image processing. [PubMed: 15460277]

2. Jones T, Carpenter A, Golland P. Voronoi-based segmentation of cells on image manifolds. Comput Vis Biomed Image Appl 2005:535–543.

3•. Unnikrishnan R, Pantofaru C, Hebert M. Toward objective evaluation of image segmentation algorithms. IEEE Trans Pattern Anal Mach Intell 2007;29:929. Adapts the clustering-based Rand index for use as a metric of image segmentation quality. [PubMed: 17431294]

4••. Jain V, Bollmann B, Richardson M, Berger D, Helmstaedter M, Briggman K, Denk W, Bowden J, Mendenhall J, Abraham W, et al. Boundary learning by optimization with topological constraints. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2010 Introduces a new metric of segmentation quality, the warping error, and a learning algorithm that optimizes it for segmentation of EM data.

[o] It is possible to write down a very simple Markov random field model that utilizes an unlimited field of view. But this model would use a single mechanism to propagate information over both short and long distances, and therefore cannot cope with multiple types of contextual information at different length scales.

HHMI Author Manuscript HHMI Author Manuscript HHMI Author Manuscript

5••. Turaga SC, Briggman KL, Helmstaedter M, Denk W, Seung HS. Maximin affinity learning of image segmentation. NIPS. 2009 Introduces a learning algorithm that directly optimizes a measure of segmentation quality (the Rand index) and applies it to EM segmentation.

6. Forsyth, DA.; Ponce, J. Prentice Hall Professional Technical Reference. 2002. Computer Vision: A Modern Approach.

7. Rosenfeld A. Picture processing by computer. ACM Comput Surv 1969;1:147–176.

8. Sobel I, Levinthal C, Macagno ER. Special techniques for the automatic computer reconstruction of neuronal structures. Annu Rev Biophys Bioeng 1980;9:347–362. [PubMed: 7396437]

9. Harris K, Perry E, Bourne J, Feinberg M, Ostroff L, Hurlburt J. Uniform serial sectioning for transmission electron microscopy. J Neurosci 2006;26:12101. [PubMed: 17122034]

10. Sporns O, Tononi G, Kotter R. The human connectome: a structural description of the human brain. PLoS Comput Biol 2005;1:e42. [PubMed: 16201007]

11. White JG, Southgate E, Thomson JN, Brenner S. The structure of the nervous system of the nematode *Caenorhabditis elegans*. Philos Trans R Soc Lond B Biol Sci 1986;314:1.

12. Denk W, Horstmann H. Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. PLoS Biol 2004;2:e329. [PubMed: 15514700]

13. Briggman KL, Denk W. Towards neural circuit reconstruction with volume electron microscopy techniques. Curr Opin Neurobiol 2006;16:562–570. [PubMed: 16962767]

14. Hayworth KJ, Kasthuri N, Schalek R, Lichtman JW. Automating the collection of ultrathin serial sections for large volume TEM reconstructions. Microsc Microanal 2006;12:86–87.

15. Smith SJ. Circuit reconstruction tools today. Curr Opin Neurobiol 2007;17:601–608. [PubMed: 18082394]

16. Knott G, Marchman H, Wall D, Lich B. Serial section scanning electron microscopy of adult brain tissue using focused ion beam milling. J Neurosci 2008;28:2959. [PubMed: 18353998]

17. Helmstaedter M, Briggman KL, Denk W. 3D structural imaging of the brain with photons and electrons. Curr Opin Neurobiol 2008;18:633–641. [PubMed: 19361979]

18. Seung HS. Reading the book of memory: sparse sampling versus dense mapping of connectomes. Neuron 2009;62:17–29. [PubMed: 19376064]

19. Lichtman JW, Sanes JR. Ome sweet ome: what can the genome tell us about the connectome? Curr Opin Neurobiol 2008;18 :346–353. [PubMed: 18801435]

20. Sobel, I.; Feldman, G. A 3 × 3 isotropic gradient operator for image processing. Presentation for Stanford Artificial Project; 1968.

21. Marr D, Hildreth E. Theory of edge detection. Proc R Soc Lond Ser B Biol Sci 1980;207:187–217. [PubMed: 6102765]

22. Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 1986:679–698.

23. Parent P, Zucker SW. Trace inference, curvature consistency, and curve detection. IEEE Trans Pattern Anal Mach Intell 1989;11 :823–839.

24. Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion. IEEE Trans Pattern Anal Mach Intell 1990;12:629–639.

25. Geman S, Geman D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans Pattern Anal Mach Intell 1984;6:721–741.

26. Li, S. Markov Random Field Models in Computer Vision. Springer; 1994.

27. Boykov Y, Veksler O, Zabih R. Fast approximate energy minimization via graph cuts. IEEE Trans Pattern Anal Mach Intell 2001;23:1222–1239.

28. Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. Int J Comput Vis 1988;1:321–331.

29. Osher S, Fedkiw RP. Level set methods: an overview and some recent results. J Comput Phys 2001;169:463–502.

30. Angelini, E.; Jin, Y.; Laine, A. Handbook of Medical Image Analysis: Advanced Segmentation and Registration Models. Springer; 2005. State of the art of level set methods in segmentation and registration of medical imaging modalities.

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

31. Tasdizen, T.; Whitaker, R.; Marc, R.; Jones, B. International Conference on Image Processing. Vol. 2. 2005. Enhancement of cell boundaries in transmission electron microscopy images; p. 129NIH Public Access

32. Yang, HF.; Choe, Y. Cell tracking and segmentation in electron microscopy images using graph cuts. IEEE International Symposium on Biomedical Imaging; 2009.

33•. Kaynig, V.; Fuchs, T.; Buhmann, JM. Neuron geometry extraction by perceptual grouping in sstem images. IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2010. Uses a graph-cut technique to solve a cost function that integrates perceptual grouping constraints with output of a random forest boundary labeling classifier

34. Macke JH, Maack N, Gupta R, Denk W, Scholkopf B, Borst A. Contour-propagation algorithms for semi-automated reconstruction of neural processes. J Neurosci Methods 2008;167:349–357. [PubMed: 17870180]

35•. Vazquez-Reina, A.; Miller, E.; Pfister, H. Multiphase geometric couplings for the segmentation of neural processes. IEEE Computer Society Conference on Computer Vision and Pattern Recognition; Los Alamitos, CA, USA: IEEE Computer Society; p. 20092020-2027.Introduces a multiphase level set scheme for EM segmentation

36. Jeong WK, Beyer J, Hadwiger M, Vazquez A, Pfister H, Whitaker RT. Scalable and interactive segmentation and visualization of neural processes in EM datasets. IEEE Trans Vis Comput Graph 2009;15:1505–1514. [PubMed: 19834227]

37. Jeong W-K, Beyer J, Hadwiger M, Blue R, Law C, Vazquez-Reina A, Reid RC, Lichtman J, Pfister H. Ssecrett and neurotrace: interactive visualization and analysis tools for large-scale neuroscience data sets. IEEE Comput Graph Appl 2010;30:58–70. [PubMed: 20650718]

38. Carlbom I, Terzopoulos D, Harris K. Computer-assisted registration, segmentation and 3d reconstruction from images of neuronal tissue sections. IEEE Trans Med Imaging 1994;13:351–362. [PubMed: 18218511]

39. Vazquez, L.; Sapiro, G.; Randall, G. Segmenting neurons in electronic microscopy via geometric tracing. International Conference on Image Processing; October: 1998; p. 814-818.

40. Bertalmío M, Sapiro G, Randall G. Morphing active contours. IEEE Trans Pattern Anal Mach Intell 2000;22:737.

41. Jurrus, E.; Hardy, M.; Tasdizen, T.; Fletcher, PT.; Koshevoy, P.; Chien, C-B.; Denk, W.; Whitaker, R. Axon tracking in serial block-face scanning electron microscopy; Med Image Anal. 2009. p. 180-188.includes Special Section on Medical Image Analysis on the 2006 Workshop Microscopic Image Analysis with Applications in Biology [Online]. Available: http://www.sciencedirect.com/science/article/B6W6Y-4SNWW88-1/2/1c7c20bc236c663b953f88c1ff67414e

42••. Andres, B.; Koethe, U.; Helmstaedter, M.; Denk, W.; Hamprecht, F. Segmentation of SBFSEM volume data of neural tissue by hierarchical classification. Proceedings of the 30th DAGM Symposium on Pattern Recognition; Springer; 2008. p. 142-152.Introduces the use of random forests for automated merging of super-voxels

43. Vincent L, Soille P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans Pattern Anal Mach Intell 1991;13:583–598.

44••. Turaga, SC.; Murray, JF.; Jain, V.; Roth, F.; Helmstaedter, M.; Briggman, K.; Denk, W.; Seung, HS. Convolutional networks can learn to generate affinity graphs for image segmentation; Neural Comput. 2010. p. 511-538.[Online]. Available: http://www.mitpressjournals.org/doi/abs/10.1162/neco.2009.10-08-881. Introduces the use of affinity graphs in EM segmentation

45. Fowlkes, C.; Martin, D.; Malik, J. Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003); 2003.

46•. Jain, V.; Murray, JF.; Roth, F.; Turaga, SC.; Zhigulin, V.; Briggman, KL.; Helmstaedter, MN.; Denk, W.; Seung, HS. Supervised learning of image restoration with convolutional networks. IEEE International Conference on Computer Vision; p. 1-8.Introduces novel convolutional networks architectures for restoration and segmentation of electron microscopy images

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

47. Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 2000;22:888–905.

48. Felzenszwalb PF, Huttenlocher DP. Efficient graph-based image segmentation. Int J Comput Vis 2004;59:167–181.

49. Cousty J, Bertrand G, Najman L, Couprie M. Watershed cuts: minimum spanning forests and the drop of water principle. IEEE Trans Pattern Anal Mach Intell 2008:1362–1374.

50•. Cardona, A. Segmented ssTEM Stack of Neural Tissue. 2010. [Online]. Available: http://www.ini.uzh.ch/acardona/data.html. First publicly available electron microscopy dataset that includes ground truth human segmentation

51. Fiala J. Reconstruct: a free editor for serial section microscopy. J Microsc 2005;218:52–61. [PubMed: 15817063]

52. Cardona, A. TrakEM2: an ImageJ-based program for morphological data mining and 3d modeling. Proceedings of the Image J User and Developer Conference; 2006.

53. Yushkevich PA, Piven J, Hazlett HC, Smith RG, Ho S, Gee JC, Gerig G. User–guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. Neuroimage. 2006

54••. Mishchenko Y. Automation of 3D reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs. J Neurosci Methods 2009;176:276–289. Introduces the first complete pipeline for EM reconstruction that incorporates computer segmentations, and claims a speedup relative to purely manual tracing. [PubMed: 18834903]

55. Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. From contours to regions: an empirical evaluation. IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2009. p. 2294-2301.

56. Kong TY, Rosenfeld A. Digital topology: introduction and survey. Comput Vis Graph Image Process 1989;48:357–393.

57. Rand WM. Objective criteria for the evaluation of clustering methods. J Am Statist Assoc 1971;66:846–850.

58•. Dollar, P.; Tu, Z.; Belongie, S. Supervised learning of edges and object boundaries. IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2006. p. 1964-1971.Introduces the Boosted Edge Learning (BEL) algorithm for machine of learning of boundary detection

59. Venkataraju, KU.; Paiva, A.; Jurrus, E.; Tasdizen, T. Automatic markup of neural cell membranes using boosted decision stumps. IEEE International Symposium on Biomedical Imaging: From Nano to Macro; 2009. p. 1039-1042.ISBI'09; 28 2009 to July 1 2009: 2009

60•. Vitaladevuni, SN.; Basri, R. Co-clustering of image segments using convex optimization applied to EM neuronal reconstruction. IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2010. Clusters supervoxels for EM segmentation using convex optimization techniques

61. Jurrus E, Paiva AR, Watanabe S, Anderson JR, Jones BW, Whitaker RT, Jorgensen EM, Marc RE, Tasdizen T. Detection of neuron membranes in electron microscopy images using a serial neural network architecture. Med Image Anal. 2010

62. Davis S, Mermelstein P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans Acoustics Speech Signal Process 1980;28:357–366.

63. Jurafsky, D.; Martin, JH.; Kehler, A. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. MIT Press; 2000.

64. Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis 2004;60:91–110.

65. Kumar, R.; Vazquez-Reina, A.; Pfister, H. Radon-like features and their application to connectomics. IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA); 2010.

66•. Veeraraghavan, A.; Genkin, A.; Vitaladevuni, S.; Scheffer, L.; Xu, S.; Hess, H.; Fetter, R.; Cantoni, M.; Knott, G.; Chklovskii, D. Increasing depth resolution of electron microscopy of neural circuits using sparse tomographic reconstruction. IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2010. Introduces an approach towards overcoming the vertical resolution limitations of TEM

HHMI Author Manuscript HHMI Author Manuscript HHMI Author Manuscript

67. LeCun Y, Muller U, Ben J, Cosatto E, Flepp B. Off-road obstacle avoidance through end-to-end learning. Adv Neural Inform Process Syst 2006;18:739.

68. Kreutz-Delgado K, Murray JF, Rao BD, Engan K, Lee TW, Sejnowski TJ. Dictionary learning algorithms for sparse representation. Neural Comput 2003;15:349–396. [PubMed: 12590811]

69. Hinton G, Salakhutdinov R. Reducing the dimensionality of data with neural networks. Science 2006;313:504. [PubMed: 16873662]

70. Marc'Aurelio Ranzato Y, Boureau L, LeCun Y. Sparse feature learning for deep belief networks. Adv Neural Inform Process Syst 2007;20:1185–1192.

71. Lee, H.; Grosse, R.; Ranganath, R.; Ng, AY. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. Proceedings of the 26th Annual International Conference on Machine Learning; ACM; 2009. p. 609-616.

72•. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86:2278–2324. Introduces convolutional networks trained by gradient learning for visual object recognition.

73. Mutch, J.; Knoblich, U.; Poggio, T. CNS: A GPU-based Framework for Simulating Cortically-organized Networks. Massachussetts Institute of Technology; 2010. Tech. Rep

74. Raddick J, Lintott C, Schawinski K, Thomas D, Nichol R, Andreescu D, Bamford S, Land K, Murray P, Slosar A, et al. Galaxy Zoo: an experiment in public science participation. Bull Am Astron Soc 2007;38:892.

75. Von Ahn L, Maurer B, McMillen C, Abraham D, Blum M. recaptcha: human-based character recognition via web security measures. Science 2008;321:1465. [PubMed: 18703711]

76. Ren, X.; Malik, J. Learning a classification model for segmentation. Proceedings of the Ninth IEEE International Conference on Computer Vision; IEEE Computer Society; 2003. p. 10

77. Hoiem D, Efros A, Hebert M. Automatic photo pop-up. ACM Trans Graph 2005;24:584.

78. He X, Zemel R, Ray D. Learning and incorporating top-down cues in image segmentation. Comput Vis ECCV 2006:338–351.

79. Gu, C.; Lim, J.; Arbeláez, P.; Malik, J. Recognition using regions. IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2009.

80. Lim, JJ.; Arbeláez, P.; Gu, C.; Malik, J. Context by region ancestry. IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2010.

81•. Jurrus, E.; Whitaker, R.; Jones, BW.; Marc, R.; Tasdizen, T. An optimal-path approach for neural circuit reconstruction. 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro; 2008. ISBI2008; May: 2008:1609–1612. 2d supervoxels are linked across sections using an optimal-path formulation

82. Yang, H-F.; Choe, Y. 3D volume extraction of densely packed cells in EM data stack by forward and backward graph cuts. Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Multimedia Signal and Vision Processing; 2009. p. 47-52.

83. Ren, X. Multi-scale improves boundary detection in natural images. Proceedings of the 10th European Conference on Computer Vision: Part III; Springer-Verlag; 2008. p. 533-545.

84. Mishchenko Y, Hu T, Spacek J, Mendenhall J, Harris KM, Chklovskii DB. Ultrastructural analysis of hippocampal neuropil from the connectomics perspective. Neuron. in press.

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

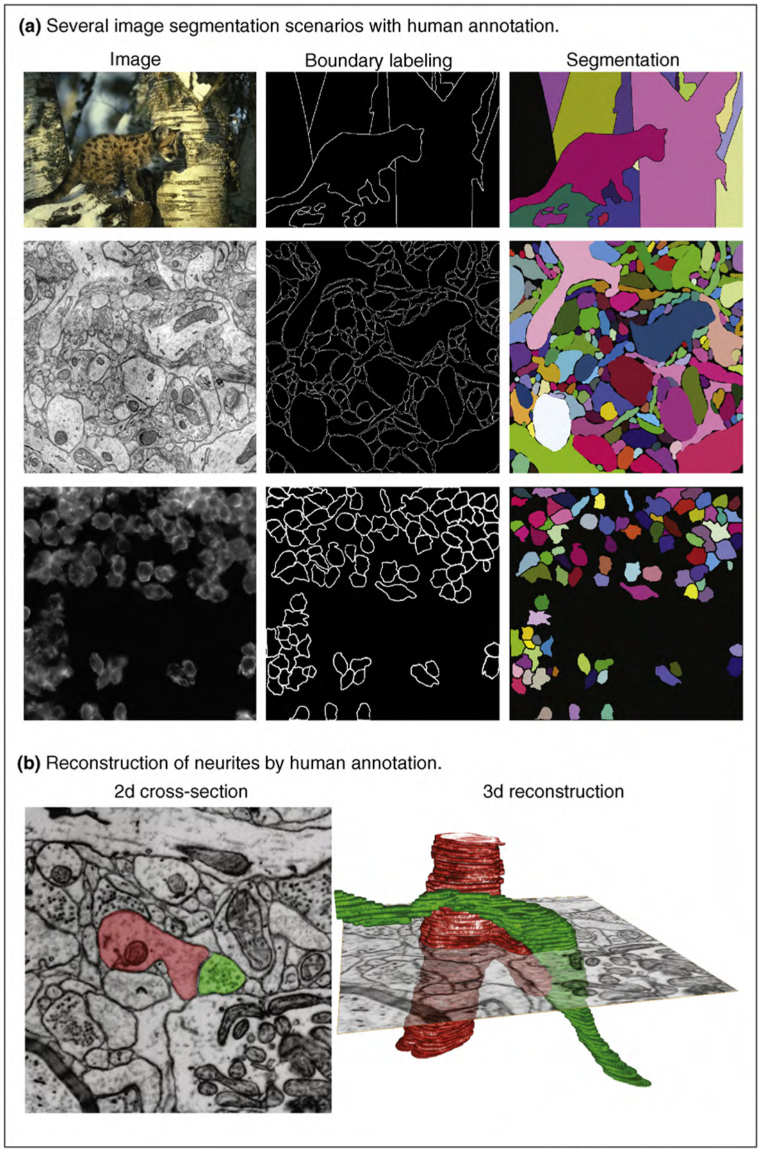
(a) Several image segmentation scenarios with human annotation.
Image
Boundary labeling
Segmentation
(b) Reconstruction of neurites by human annotation.
2d cross-section
3d reconstruction

Figure 1.

Segmentation of various types of images. **(a)** Natural image from the Berkeley Segmentation Dataset [1•], scanning EM image of mouse cortical neurons (image courtesy of the Lichtman lab, annotation by Daniel Berger), fluorescent image of actin-labeled *Drosophila melanogaster* cells [2]. In all cases, segmentation has been provided by human annotation. **(b)** Cross-sections of a dendrite (red) and axon (green) in an EM image of hippocampus, colored by hand using ITK-SNAP software, and 3d reconstructions of the axon making a synapse onto a spine of the dendrite, assembled from EM images of a series of 30 nm slices. (image courtesy of the Lichtman lab, and annotation by Daniel Berger).

HHMI Author Manuscript HHMI Author Manuscript HHMI Author Manuscript

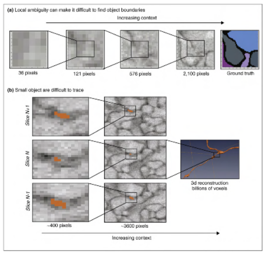

Figure 2.

Context and ambiguity in neurite tracing. **(a)** Larger context is needed to identify an object boundary due to a ‘gap’ in an image boundary. **(b)** Larger context is needed to trace an object through an area where it becomes very thin. There is also context in the third dimension, which is not shown here. (Original image courtesy of Kevin Briggman, Moritz Helmstaedter, and Winfried Denk.)

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

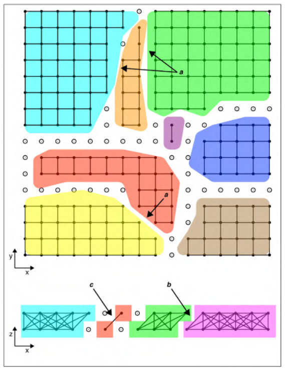

Figure 3.

Affinity graphs are a more powerful output representation than boundary labelings. The nodes of the graph represent pixel locations, while edges between nodes indicate affinities. Boundary pixels are represented as circled nodes lacking edges. The graph allows for objects to be separated by boundaries that are less than a pixel wide as in regions marked *a* in the *x*–*y* plane or *b* in the *x*–*z* plane. Objects that would have been split due to misalignment of 2d image slices can be connected by graph edges spanning different image slices as in *c*. Affinity graphs are shown here for two-dimensional images, but can also be defined in three or higher dimensions.

HHMI Author Manuscript

HHMI Author Manuscript

HHMI Author Manuscript

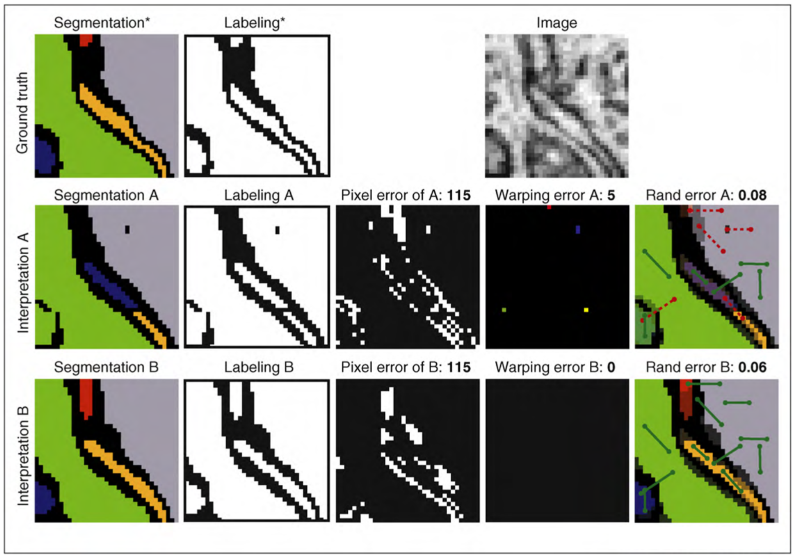

Figure 4.

Three measures of segmentation quality. A ground truth segmentation can be used to quantify the performance of different segmentation algorithms (A and B). Machine learning algorithms require such error metrics in order to automatically optimize their parameters to improve performance. *Pixel error* measures the number of misclassified pixels in the boundaries of two segmentations. While simple to compute, it classifies pixels as errors even when there is no real change in the segmentation. Segmentations A and B have the same pixel error. *Warping error* penalizes misclassified pixels that lead to topological errors (red = deletion, green = merger, yellow = split, and blue = hole in Segmentation A). Segmentation A has more warping error than Segmentation B. *Rand error* is a clustering measure that counts the number of pixel pairs that are incorrectly grouped. Segmentation A has more Rand error than Segmentation B. The Berkeley metric (not illustrated) introduced in [1] assigns Interpretation A *lower error* than Interpretation B (see [4] for details).We acknowledge support from the Howard Hughes Medical Institute and the Gatsby Foundation. We are grateful to D Berger, A Cardona, D Chklovskii, Y Choe, W Denk, S Emmons, F Hamprecht, M Helmstaedter, L Jurrus, Y LeCun, J Macke, and A Vazquez-Reina for comments and corrections.