APPLICATIONS OF BEAMLETS TO DETECTION AND EXTRACTION OF LINES, CURVES AND OBJECTS IN VERY NOISY IMAGES

David Donoho

Stanford University, Dept. of Statistics
Stanford, CA 94305
donoho@stat.stanford.edu

Xiaoming Huo

Georgia Inst. of Tech., School of ISyE
Atlanta, GA 30332
xiaoming@isye.gatech.edu

ABSTRACT

Beamlets are a special dyadically organized collection of line segments, exhibiting a range of lengths, positions and orientations. This collection is relatively compact: there are $O(n^2 \log_2(n))$ beamlets, compared to $O(n^4)$ line segments. This collection is relatively expressive: up to a certain tolerance, it does not take more than $O(\log_2(n))$ beamlets to approximate a single line segment. Because of these two properties, chains of a relatively few beamlets can build quite general curves. The beamlet pyramid is a multiscale data structure which stores all the line integrals of the image over all beamlets. By summing a relatively few coefficients from the beamlet pyramid, one can obtain integrals of the image along quite general polygonal curves.

We consider the beamlet pyramid a natural platform on which to build new methods for detecting linear and curvilinear features in very noisy data. The general approach is to solve such problems by adaptively constructing chains of beamlets which extremize certain integrals over the image. We give examples in the problems of detecting the presence of line segments in very noisy data; detecting curves in noisy data; and in the problem of extracting objects in very noisy data. We are able, in examples, to detect objects which seem practically invisible to the unaided eye.

1. INTRODUCTION

Identifying linear and curvilinear image features is a common problem in applications of image processing. There is an overwhelmingly large literature on methods of detecting such features, for example under the name 'edge detection', so a comprehensive summary is beyond hope, but it is possible to make a few generalizations.

- Classical 'edge detectors' seem to work well when the noise is nonexistent or weak.
- Classical detectors work only at the finest scale of the image – detecting edges based on behavior of the image in a sliding window a few pixels wide.

We are interested in a situation where an underlying linear or curvilinear feature is embedded in a very noisy image. In fact, we are interested in settings where the pixel signal-to-noise ratio, computed in a pixel centered on the feature, is very small. In such a setting, essentially by definition, an edge detector based on summing over a window a few pixels wide will not work, as we are assuming that the signal-to-noise ratio is very poor, and this will remain the case for statistics based on averages over a few pixels, whatever be the details of the averaging.

This work was partially supported by NSF grants DMS 98–72890 and NSF ECS-9707111 and by AFOSR MURI-95-P49620-96-1-0028, and by Georgia Tech start-up funds for Dr. Huo

It might seem intuitively plausible that, if the linear or curvilinear feature is extended across a substantial range, then averaging will help in some way. This suggests that using an edge detector based on a local filter which linearly combines values within a rather large block of pixels will provide a better signal-to-noise ratio. However, this intuition is misguided. For a linear feature a few pixels wide, a sum over a block of side m by m will have m pixels on the feature, out of m^2 total. Since the noise in this average scales as $\sqrt{\#(\text{terms in average})} \approx m$ it follows that the signal-to-noise ratio of local filtering will remain about the same no matter what the size of the window. In short, the problem with existing edge detectors in a very noisy environment is not the fact that they operate at pixel-scale, because they would continue to fail when operated at any fixed scale.

In this paper we describe a set of tools which allow us to develop the equivalent of an adaptive filtering highly matched to the problem of finding lines and curvilinear features in images. We give examples showing that we can successfully identify features in extremely heavy noise – in some cases, noise which is so heavy that the object to be detected is nearly invisible to the unaided eye.

Our methods are built on a multiscale data structure, the *beamlet pyramid*. Beamlets are a special collection of dyadically organized line segments at all scales, locations and orientations. The beamlet pyramid is the collection of all line integrals of the image data along line segments defined by the beamlet pyramid. By chaining together small numbers of beamlets, one can obtain any line segment, and can also obtain a wide range of polygons. Correspondingly, by summing together a small number of coefficients in the beamlet pyramid, one can obtain any line integral, and a wide range of curvilinear integrals over polygons.

We discuss in this paper some algorithms which work with the beamlet pyramid to detect lines, curves and objects. In some sense they work by adaptively chaining together beamlets into line segments or polygons, according to an optimization principle.

The application areas for our algorithms are as follows:

1. *Line Detection.* Suppose we are give a noisy picture, which might potentially contain within it a linear feature against a constant background. The feature, if present, is very weak. How to test for its presence? A classical approach would use the *Generalized Likelihood Ratio Test* (GLRT). The idea of GLRT is as follows: At each time, we test a pair of hypotheses. One (the alternative) claims the presence of a specific line segment. The other (the null) claims that there is

no feature at all. We compare the two hypotheses by considering their likelihood ratio. We repeat this for all possible line segments. The line segment that gives the largest likelihood ratio defines the GLRT statistic.

Apparently, to run the GLRT we need to test at least $O(n^4)$ hypotheses. Without even considering the effort of calculating each likelihood ratio, the complexity of the algorithm is at least $O(n^4)$. We can apply instead a principle of adaptive chaining in the beamlet dictionary. Working only among beamlets which yield a sufficient likelihood ratio, we build straight chains starting from each such beamlet searching for chains with higher statistics. This can yield an approximation to the GLRT $O(n^2 \log_2(n))$ algorithm. So beamlets provide a low-complexity algorithm to obtain an essentially optimal detector.

2. *Curve Detection.* Think of an image as associated to a graph structure: the pixels are associated with vertices, and the near-neighbor vertices are connected by edges. A path is a chain of edges, which connects two given vertices in an noisy picture. We are interested in detecting whether the image is only noise, or whether there is, in addition to noise, a polygonal feature supported on a path connecting the two vertices.

   We might consider using the idea of GLRT. However, as the size of the image n increases, the number of paths between two vertices grows exponentially. A direct application of GLRT would end up with an intractable optimization problem even when the image size n is 64. Also, it is highly unclear whether a GLRT searching through an amazingly large number of curves is not simply going to 'find' structure in pure random noise.

   We consider instead an algorithm oriented around the beamlet graph, i.e. the graph associated with an image where two pixels are connected not only if they are nearest neighbors, but also if there are beamlets beginning in one pixel and ending in the other. We consider an estimator based on optimizing, over paths, a ratio of a complexity-penalized curvilinear integral along the path to the length of the path. We may solve the problem for modest image size n using a heuristic path-search procedure. Results given below show surprising effectiveness in extremely noisy environments.

3. *Object Extraction.* We study the problem of estimating an isolated object in a very noisy picture. The object is nonzero throughout a region in the image, over which the image has a constant, small amplitude, whereas outside the object the image is zero. Very heavy noise is added.

   We find a region which approximately solves the problem of optimizing the ratio of the integral over the region to the square root area of the region. In detail, we exactly solve the problem of optimizing the ratio I/L where I is a complexity-penalized measure of the integral over the region and L – a proxy for the square root of the area of a region – is the length of the region's boundary. Using the network flow literature [1], this type of problem is called the *minimum cost to time ratio cycle* (MCTTRC) problem, and can be solved as a linear programming (LP) problem; we can use either interior point methods or simplex methods to solve it. Results given below show surprising effectiveness in extremely noisy environments.

Due to space limitations, we omit many significant details. In the rest of this paper, we briefly summarize only the most important concepts, and then provide pointers to the existing literature where possible. We focus on describing three simulations that demonstrate our major claims.

2. BEAMLETS, A PRIMER

Beamlets provide us an organizational tool which is very useful in searching for linear and curvilinear structures in image data. There are five interrelated concepts to: beamlet dictionary, beamlet transform, beamlet pyramid, fast beamlet transform, and beamlet graph. Some of these ideas, or near relatives, have appeared before, without special names or under other names. We mention some prior literature.

2.1. Beamlet Dictionary

Beamlets make up a special subcollection of line segments associated with an n by n image. They are generated following three steps: (1) exhaustive dyadic recursive partitioning of the unit square, (2) marking equispaced vertices along the boundary of every resulting dyadic subsquare, and (3) connecting all pairs of vertices within each square by a line segment. A description, together with a graphical illustration can be found in [5], compare also [7], where the same collection appeared by a different name. The line segments that result are called beamlets and the collection of all such beamlets is the *beamlet dictionary*. Although there are $O(n^4)$ line segments associated with vertices on an n by n grid, there are only $O(n^2 \log(n))$ beamlets. Nevertheless, it takes only $O(log(n))$ beamlets to build an approximation to any line segment, valid to within one pixel.

2.2. Beamlet Transform

The *beamlet transform* is, loosely speaking, the collection of all line integrals of the image defined by line segments in the beamlet dictionary.

More precisely, given a digital image, we first perform a piecewise-constant interpolation, creating a function on the continuum which is constant on each pixel. Then we associate to each given beamlet a *beamlet coefficient* defined by integration of the interpolated function along this beamlet. The beamlet pyramid is the data structure comprising all such beamlet coefficients.

2.3. Two-scale Relationship

Beamlet dictionary obeys a certain 2-scale relationship: each beamlet at a coarser scale is approximately the union of at most three beamlets at the next finer scale.

More precisely, consider a dyadic square, the parent, and its four dyadic children; and consider a beamlet associated with the parent. We can see by inspection that this coarse-scale beamlet can be broken into 1, 2, or 3 disjoint line segments associated to the child dyadic squares which are traversed by the line segment. Those shorter segments can all be approximated to within one pixel distance by beamlets associated to the children squares.

This 2-scale relationship of beamlets can be extended to beamlet coefficients: a coarse scale beamlet coefficient can be approximated by the summation of a few fine-scale beamlet coefficients.

In the wavelet literature, a "pyramid" is used to describe data structures where the fine-scale data may be used to calculate coarse-scale data. So we call the beamlet coefficient structure a *beamlet pyramid*.

2.4. Fast Algorithms & Hough Transform

The 2-scale relation we have just mentioned has been discovered and applied within the context of fast algorithms for Hough transforms and Radon transforms, see [9, 4, 3]. For example, Goetze and Druckmiller [9] defined essentially the same data structure as beamlets (without the beamlet name, and with other applications in mind) and used a bottom-up recursive algorithm based on the two scale relationship to approximately calculate all integrals over lines traversing the image from one boundary pixel corner to another, in $O(n^2 \log(n))$ time. During the execution of this algorithm, beamlet coefficients are calculated at intermediate steps. Brandt and Dym [4], without defining specifically the beamlet dictionary, also pointed out a two scale relationship to calculate approximate multiscale line integrals of an image over an extensive collection of line segments in the image.

What is different here, besides the introduction of terminology? In our software package, we typically do not use the 2-scale relationship to perform fast approximate integration, preferring exact integration according to a slower algorithm. This is because we wish to introduce the beamlet pyramid as an important data structure, on which certain specific graph-theoretic algorithms can be built. For this purpose, we appear to need accuracy; see the related paper [7], which used the beamlet pyramid (by another name) and found that very high (in fact, sub-pixel) accuracy was required.

2.5. Beamlet Graph

We can associate a multiscale *beamlet graph* to an n by n grid, as follows. Consider all the corners of all the pixels in an image. These form a $n+1$ by $n+1$ array of points. We consider each point in this array as a vertex in the beamlet graph. If two points are connected by a beamlet, then there is an edge between the two corresponding vertices in the beamlet graph.

This is a multiscale graph, containing some edges whose associated line segments traverse only a few pixels; other edges have line segments which traverse order n pixels. It has very inhomogeneous connectivity, with some vertices connected to only 8 other vertices, while others are connected to about $4n$ other vertices. The average connectivity is to about $\log(n)$ other vertices.

This graph is useful as the foundation for a number of interesting algorithms which build paths through the graph structure automatically, in the solution of some detection or estimation problem.

3. LINE DETECTION

We have explained the general idea of line detection in the Introduction. Here we briefly describe one computational experiment, and emphasize why this simulation verifies our major claim. The main conclusion is that, by using beamlets, we can have a low complexity algorithm yielding a good approximation to the GLRT.

Figure 1 shows the results of an experiment where a very faint line segment is embedded in a noisy array, the pixel signal-to-noise ratio varying from -30 to -1. Obviously, no pixel-level analysis is going to detect this feature. The graph gives median values of maximum score of the classical GLRT based on the collection of line segments, and a threshold for the GLRT, derived using intimate knowledge of the beamlet statistics. Also depicted is the median maximum standardized beamlet coefficient from the beamlet transform. Both methods are able to reject the null hypothesis of no line segment when the they exceed threshold. Both methods start to work when the SNR is around -24. In contrast, pixel-based detectors will not be able to tell whether there is an underlying object until the SNR is nearly -10.

Evidently, the largest standardized beamlet coefficient is almost as large as the GLRT, and so it is almost as good a detector. Yet the largest beamlet coefficient can be computed approximately using an algorithm of order $O(n^2 \log^2(n))$ as compared to $O(n^5)$ for the GLRT.

It is possible to improve on the maximum standardized beamlet coefficient. We have developed an algorithm which grows chains starting from 'seeds' provided by the $\sqrt{n}$ highest standardized beamlet coefficients, and grows chains which enumerate all approximations to straight line segments having the given seed as its maximal-length beamlet coefficient. The resulting algorithm provides an ε-approximation to the full GLRT with scarcely more work (in asymptotic order) than simply finding the maximum standardized beamlet coefficient.

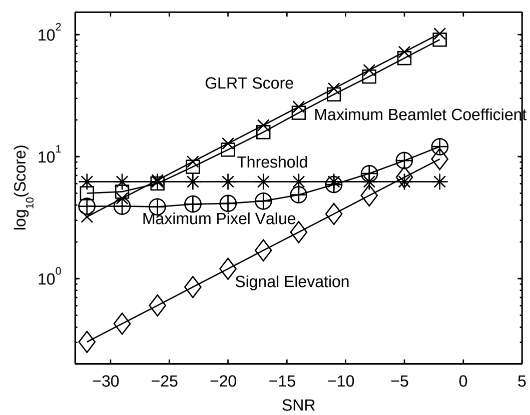

Fig. 1. Some quantities associated with beamlet-based line detection. We consider eleven different SNRs. The line with $\diamond$ indicates the pixel S/N of the underlying line object. The line with $\oplus$ indicates the median maximum of pixel intensities. The line with $*$ indicates the threshold of detectability. The line with $\square$ indicates the median maximum of the standardized beamlet coefficients. The line with $\times$ indicates the median maximal scores of GLRT based on line segments. All the median maximal values are based on 100 simulations. The size of the test image n is 128.

4. CURVE DETECTION

In Figure 2, we summarize an experiment in a beamlet-based curve detection. We embed a very weak signal in a very strong noise. The signal should be the indicator function of a filament and should be so weak that, roughly speaking, it is indetectable to the unaided eye. As one can see, the algorithm we develop is able to recover a surprisingly accurate reconstruction of the filament.

Our curve detection procedure is based on the use of network flow algorithms in the beamlet graph. The general goal is to find a path through the beamlet graph optimizing a certain functional.

This path corresponds to a polygonal chain of beamlets in the image, and the functional we optimize is the ratio of the integral along the chain to the arclength of the chain.

This ratio-optimization problem falls under the scope of so-called minimum cost-to-time ratio problems, and we have developed a fast heuristic procedure for obtaining a reasonable solution, based on ideas from dynamic programming. We run a small number of iterations of a basic step which attempts to find a best path from p to p', say, by using the best path known at the previous iteration to get from p to any beamlet graph neighbor of p' and to compose that path with the beamlet from the neighbor to p'.

We have also found it useful to modify the integral-to-length ratio to take into account the noise in the data, and have developed a statistical complexity penalization, where we penalize polygonal paths proportional to the number of links that they contain. This discourages the finding of spurious filaments due to the influence of noise on the optimization process.

In Figure 2, we show some results of simulations, testing the two variants mentioned above. The top figure is the original curve. The second row contains six very noisy pictures with the above curve embedded very faintly. The third row contains results by optimizing a complexity penalized ratio. The fourth row contains results based on optimizing a ratio without penalization. In experiments 1, 3, and 4, penalization seems to impose better local structure. All simulation results are based on 20 iterations of the basic propagation step.

Comparing to the near-neighbor graph, we have two advantages of working on the beamlet graph. The first advantage is an algorithmic one: in the beamlet graph, the representation of a polygonal path is relatively simple, hence it takes fewer iteration steps for the algorithm to find the final solution. The second advantage is a statistical one: by adding a penalty term based on the statistical significance of the presence of a filament, we can discourage selection of unpromising filaments, hence we may accelerate the convergence.

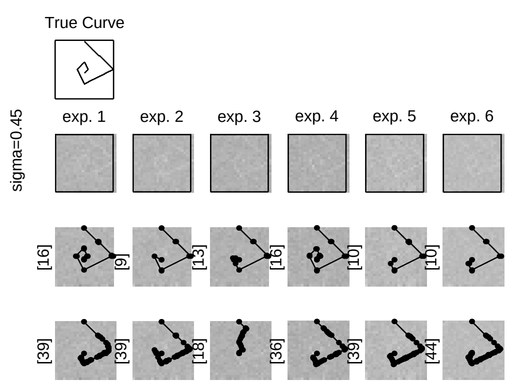

Fig. 2. Results of curve estimation. The second row shows the noisy images. The third row shows the results from minimizing a complexity penalized ratio. The fourth row shows the results from minimizing a ratio without penalization.

5. OBJECT DETECTION

Although beamlets are obviously relevant for filament detection, they are, owing to known graph-theoretic optimization algorithms, also relevant to object detection. This can be seen by adapting to the beamlet graph insights that Jermyn and Ishikawa [11] had about object extraction based on graphical algorithms in the nearest-neighbor graph. There are two key facts. First, each cycle in the beamlet graph corresponds to a polygonal chain of beamlets which is a cycle in the image and so encloses a region. Second, by applying partial integration, the vector beamlet transform of the vector image consisting of, for its y-component, the vertical-primitive of the image and for its x-component, the horizontal-primitive of the image, contains information about integrals of the image over regions. Hence, by looking for cycles which optimize a ratio of beamlet integral over length, one obtains a region which approximately optimizes the empirical detection score of GLRT. A key observation we have made in the beamlet graph case (which does not arise in the near neighbor case) is that the resulting optimization problem is a form of minimum cost-to-time-ratio cycle problem, and so a globally optimal region with boundary made of a chain of beamlets can be obtained by linear programming [1].

Figure 3 shows some numerical simulation results. The top image is the noiseless object. The second row contains noisy images at various SNR's. The third row contains results of running our linear programming algorithm with no penalty on the boundary complexity. The fourth and the fifth rows contain the results when a penalty term defined on the beamlets is added, and the penalizing parameter is 2 or 4. We observe that with adequate amount of penalty on the complexity of the boundary, we indeed obtain estimators with simpler boundaries.

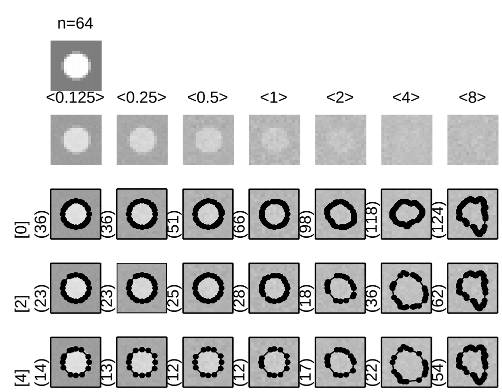

Fig. 3. Simulations on circle images. The values in arrows $<>$ are the values of σ, the noise standard deviation. The values in brackets [ ] are the values of λ, which is the penalty parameter. The values in the parentheses ( ) are the numbers of beamlets in the final solutions.

6. CONCLUSION

Beamlets provide a conceptual and software toolbox which is well-suited to detect linear features in very noisy pictures. We demonstrated some experiments in the detection of lines, curves, and objects with regular boundary. We were able to obtain success at extremely low signal-to-noise ratios. At the conference we hope to present detailed performance comparisons with existing techniques.

7. REFERENCES

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] D. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 1995.

[3] M. L. Brady. A fast discrete approximation algorithm for the radon transform. *SIAM J. Computing*, 27(1):107–19, February 1998.

[4] A. Brant and J. Dym. Fast calculation of multiple line integrals. *SIAM J. Sci. Comput.*, 20(4):1417–1429, 1999.

[5] David Donoho and Xiaoming Huo. Beamlet pyramids: A new form of multiresolution analysis, suited for extracting lines, curves, and objects from very noisy image data. In *Proceedings of SPIE*, volume 4119, July 2000.

[6] David Donoho and Xiaoming Huo. *A Hierarchy of Beamlet-based Algorithms for Multiscale Image Analysis*. Stanford U. and Georgia Inst. of Tech., March 2001.

[7] D.L. Donoho. Wedgelets: Nearly minimax estimation of edges. *Annals of Statistics*, 27(3):859–897, 1999.

[8] D. Geiger, A. Gupta, L.A. Costa, and J. Vlontzos. Dynamic programming for detecting, tracking and matching deformable contours. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(3):294–302, 1995.

[9] W.A. Götz and H.J. Druckmüller. A fast digital radon transform – an efficient means for evaluating the hough transform. *Pattern Recognition*, 28(12):1985–1992, 1995.

[10] Xiaoming Huo. *Sparse Image Representation via Combined Transforms*. PhD thesis, Stanford, August 1999.

[11] Ian Jermyn and Hiroshi Ishikawa. Globally optimal regions and boundaries. In 7*th ICCV*, Kerkyra, Greece, September 1999.

[12] Richard M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23:309–311, 1978.

[13] R.M. Karp and J.B. Orlin. Parametric shortest path algorithms with an application to cyclic staffing. *Discrete Applied Mathematics*, 3:37–45, 1981.

[14] J.-L. Starck, F. Murtagh, and A. Bijaoui. *Image Processing and Data Analysis*. Cambridge, 1998.

[15] Ofer Zeitouni, Jacob Ziv, and Neri Merhav. When is the generalized likelihood ratio test optimal? *IEEE Trans. Inform. Theory*, 38(5):1597–1602, 1992.