# Beamlet Pyramids: A New Form of Multiresolution Analysis, suited for Extracting Lines, Curves, and Objects from Very Noisy Image Data

David L. Donoho[a] and Xiaoming Huo[b]

[a]Stanford University, 370 Serra Mall, Stanford, CA 94305-4065
[b]Georgia Tech., 765 Ferst Dr., Atlanta, GA 30332-0205

July 3, 2000

## ABSTRACT

We describe a multiscale pyramid of line segments and develop algorithms which exploit that pyramid to recover image features – lines, curves, and blobs – from very noisy data.

The beamlet dictionary is a dyadically organized collection of line segments, occupying a range of dyadic locations and scales, and spanning a full range of orientations. It is an efficient substitute for the full dictionary of "beams" (the collection of all possible line segments connecting pairs of pixels in an image). The beamlets dictionary has low cardinality (there are $O(n^2 \log(n))$ beamlets as compared to $O(n^4)$ beams). Despite the reduced cardinality, it takes at most $8 \log_2(n)$ beamlets to approximate any edgel to within distance $2/n$. A wide range of polygonal curves can be built from chains of relatively few beamlets.

The *beamlet transform* of a function $f(x_1, x_2)$ is the collection of integrals of $f$ over each segment in the beamlets dictionary. The resulting information is stored in an *beamlet pyramid*. One can use this to rapidly calculate integrals of $f$ for any of a wide range of polygonal curves, simply by summing together a few selected coefficients from the pyramid.

In analyzing faint signals embedded in very noisy data, integration over substantial numbers of pixels can yield statistics with favorable signal-to-noise ratios, provided the signal is near-constant over the averaging region. When the noise-free signal has features nearly constant along linear and curvilinear structures, the beamlet pyramid of the noisy image may contain a body of statistically reliable information.

In our recent work, we have developed optimization algorithms which combine numerous coefficients from the beamlet pyramid, adaptively chaining together beamlets of a variety of scales and locations to detect/recognize curves in noisy image data. By partial integration, these algorithms also allow for object detection and recognition. In theory, the new methods are far more sensitive and accurate than any previously proposed methods of edge detection and object extraction. Empirical results with small images correlate well with the theoretical claims.

**Keywords:** Beamlets, Wedgelets, Complexity Penalized Residual Sum of Squares (CPRSS), Recursive Dyadic Partitioning (RDP), Minimum Cost to Time Ratio Path (MCTTRP), Minimum Cost to Time Ratio Cycle (MCTTRC).

Further author information: (Send correspondence to X. Huo) Xiaoming Huo: E-mail: xiaoming@isye.gatech.edu; David L. Donoho: E-mail: donoho@stat.stanford.edu.

## 1. BEAMLETS

We consider an image as a function residing on a $[0, 1] \times [0, 1]$ unit square. It is piecewise constant, with "pixels" of side $1/n$ by $1/n$. The collection of beamlets $\mathcal{B}_{n,\delta}$ is a multiscale collection of line segments occurring at a full range of orientations, positions, and scales. It is generated as follows:

1. *Dyadic Subdivision.* We form all dyadic subsquares of $[0, 1]^2$ in the obvious way; to begin we divide the unit square into four subsquares of sidelength $1/2$. Each subsquare is then divided into four smaller subsquares, and so on. Figure 1 shows some subsquares after $0, 1, 2$ or $3$ steps of subdivision. We continue until we have created all dyadic squares of side $1/n$ by $1/n$ or larger.

2. *Vertex Labelling.* For definiteness, think of $\delta$ as $1/n$, although in certain applications $\delta$ should be far smaller. Traversing the boundary of each subsquare, we mark out equally spaced vertices at spacing $\delta$. Note that the distance between neighboring vertices is $\delta$, no matter which scale subsquare we consider.

3. *Connect The Dots.* In each subsquare, form the collection of all line segments connecting any pair of vertices. Any such line segment is called a beamlet.

(Beamlets were first introduced[1] under the name "Edgelets". We now think that Beamlets is a better name because edge-finding is only one application of this concept).

Figure 1 shows four beamlets at different scales.

Let $\delta = 1/n$. Then there are $O(n^2 \log(n))$ beamlets. In comparison, the number of beams (beam = connection between arbitrary pixel corners) is much larger: $O(n^4)$. So beamlets offer a reduced cardinality substitute for beams. The beamlets dictionary, although of reduced cardinality, offers a sparse approximation to the entire collection of beams. Any beam can be approximated within Hausdorff distance $1/n$ by a chain of at most $C \log(n) + 1$ beamlets.

The beamlet transform is simply the collection of all line integrals formed by viewing the image as a piecewise constant object and integrating along each line segment in the beamlet dictionary. The data structures of all coefficients obtained in this way we call the beamlet pyramid. In fact, it truly is a pyramid, since the integrals obey a two-scale relation: data on line integrals at finer scales allows to calculate the line integrals at coarse scales.

Roughly speaking, our work revolves around three ideas:

1 The beamlet transform contains multiscale, multidirection information, some of which can be very powerful in settings of finding extended linear or curvilinear structure in high noise.

2 By piecing together short chains of beamlets, we can efficiently explore a massive collection of polygonal paths, and can potentially identify very efficiently just the paths at maximal signal-to-noise ratio.

3 By appropriately defined optimization problems, we implicitly obtain automatic methods for chaining together beamlets across a range of scales, orientations and locations.

In this paper we briefly expose this triumvirate of ideas, and hope to convince the reader that beamlets provide interesting new approaches and insights for a variety of established problems, with much unexplored potential.

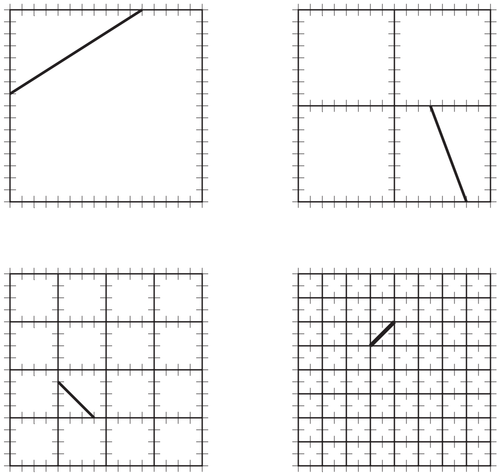

**Figure 1.** Four Beamlets, at various scales, locations, and orientations.

## 2. APPLICATION: DETECTING A LINEAR FEATURE

Beamlets allow to detect the existence of a linear feature in a noisy image. We suppose there might possibly be a beam – the indicator of a line segment – present in the image, but embedded in extremely high noise. If we look at pixel-level statistics, a large positive value would be indicative of presence of the beam. However, suppose now that the SNR is so low that none of the pixel values is likely to yield significance. Then, pixel level detection will fail. A traditional response to noise is to apply bandpass filtering, in effect averaging over substantial neighborhoods in order to suppress noise. However, any method based on bandpass filtering must also fail – the signal-to-noise ratio based on standard filter outputs will not be larger than the ratio at pixel level, no matter how big an area one averages over.

Despite the failure of pixel-level detection and of bandpass filtering, the object may still be detectable. Everything depends – by the theory of matched filtering – on what would happen if we could take the inner product of the image data with the indicator of the "true underlying beam" (which of course we would not typically know). Integrating along *exactly* the beam which is present in the image provides the ideal detector statistic. Now it is easy to see that there is a range of SNR where this ideal statistic would yield detection, while the object would be undetectable by pixel level statistics and/or bandpass filtering.

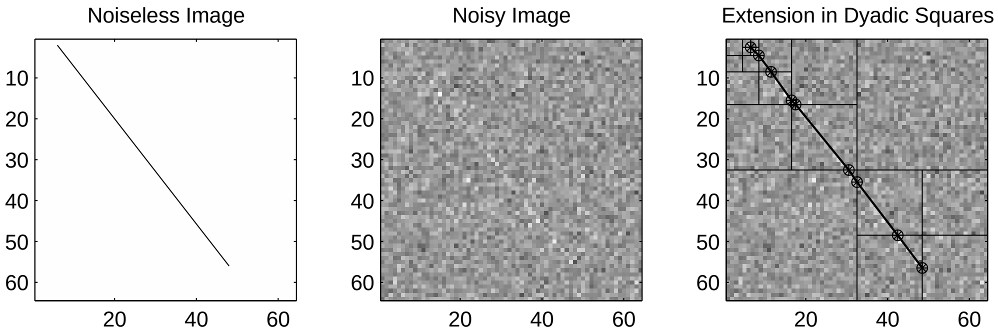

**Figure 2.** If there is a linear object embedded in a noisy image (shown on the left), we can detect it by extending beamlets in dyadic squares (shown on the right). By this extension, we can aggregate signal on a line, therefore we may obtain a better signal to noise ratio.

Typically we are searching for the presence of an unknown beam – unknown in location, length and orientation. In that setting, we must calculate all integrals over the image along all possible beams, and use the maximum normalized value to make our decision (doing this in the right way will yield the so-called *GLRT detector*). Unfortunately there are $O(n^4)$ beams and so this would require $O(n^5)$ work.

We could instead simply take the maximum normalized value in the beamlet pyramid. As the table below shows, the partial GLRT based on this strategy can yield far higher detectability in low SNR situations than could statistics based on bandpass filtering or pixels.

| SNR | Signal Ampli. | Max. Pixel Value | $2\sqrt{\log(n)}$ | Max. Beamlet Coeff. | Max. Wavelet from Scale 1 to 6 Coarser $\Longrightarrow$ Finer | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 1 | 2 | 3 | 4 | 5 | 6 |
| -13.0 | 0.614 | 3.748 | 6.230 | 6.366 | 1.114 | 1.558 | 2.342 | 2.883 | 3.238 | 3.808 |
| -12.0 | 0.774 | 3.690 | 6.230 | 6.107 | 0.973 | 2.019 | 2.153 | 2.635 | 2.829 | 3.330 |
| -11.0 | 0.974 | 4.198 | 6.230 | 9.185 | 2.446 | 3.428 | 3.067 | 3.388 | 2.598 | 3.599 |
| -10.0 | 1.226 | 4.071 | 6.230 | 10.195 | 1.668 | 1.998 | 2.488 | 3.124 | 3.425 | 4.023 |
| -9.0 | 1.543 | 4.000 | 6.230 | 13.501 | 1.625 | 2.592 | 2.367 | 3.452 | 3.218 | 3.516 |
| -8.0 | 1.943 | 4.429 | 6.230 | 15.812 | 1.445 | 2.366 | 2.294 | 2.540 | 3.321 | 4.026 |
| -7.0 | 2.446 | 4.314 | 6.230 | 19.403 | 3.295 | 3.164 | 2.875 | 2.611 | 3.471 | 6.252 |
| -6.0 | 3.080 | 5.519 | 6.230 | 26.876 | 3.442 | 2.768 | 3.876 | 4.687 | 4.474 | 4.880 |
| -5.0 | 3.877 | 6.341 | 6.230 | 33.037 | 4.575 | 5.259 | 5.919 | 5.526 | 5.596 | 5.534 |
| -4.0 | 4.881 | 6.695 | 6.230 | 40.830 | 5.382 | 5.246 | 6.551 | 5.334 | 6.668 | 7.747 |

**Table 1.** From left to right, this table lists signal to noise ratios (SNR), amplitudes of the underlying linear feature, maximum pixel values in the noisy image, ideal thresholds $2\sqrt{\log(n)}$, maximum beamlet coefficients of the noisy image, and maximum coefficients after a bandpass filter at different scales. This table illustrates that when the SNR is between $-11.0$dB and $-5.0$dB, see rows with underlined entries, any bandpass filter will not detect the linear feature, while beamlet transform will detect it. The image size is 128 by 128. The linear feature starts at $(1/32, 3/32)$ and ends at $(7/8, 6/8)$. We choose a 2-D tensor product of Daubechies 4 wavelets as a representer of bandpass filters.

There are two formal principles which motivate the partial GLRT based on the beamlet pyramid.

1 *For any beam, there is always a beamlet lying 'along' it with a length at least c times as long* where $c$ is an absolute constant. Effectively, $c = 1/8$. It follows that the maximum value of the GLRT over

the reduced cardinality beamlet pyramid is at least proportional to the GLRT over the full collection of beams.

2 The partial GLRT based on the beamlet pyramid is at most $O(n^3)$ work rather than $O(n^5)$ work for the full GLRT. Moreover, fast approximate algorithms exist which yield $O(n^2 \log(n))$ work.

A more specific example is shown in Figure 2 and Table 1. Figure 2 shows a beam in a noiseless image and an in the presence of noise (the middle one). The right sub-figure shows how the linear feature can be decomposed into beamlets. Evidently, there is a beamlet which closely tracks the beam a large proportion of its length. Hence the associated beamlet coefficient may in statistical sense be significantly large. To verify this, we show some quantities in Table 1. In the table, when the signal to noise ratio (SNR) is equal to $-11.0$dB, the maximum beamlet coefficient is above the detector threshold, which indicates that we can reliably say that there is a linear feature in the image. Meanwhile neither the maximum pixel value, nor the maximum wavelet coefficient can tell us this. When the SNR is between $-10.0$dB and $-6.0$dB, this phenomenon still holds. When the SNR is equal to $-5.0$dB, the maximum pixel value becomes larger than the detector threshold, hence the linear feature becomes significant at the pixel level. Note for wavelets, it only becomes sensitive to the existence of the underlying linear feature when the SNR is further increased to $-4.0$dB. At this level, we can in principle already see the feature from pixel values directly.

A more sophisticated detector scheme would chain together concatenations of several beamlets. Doing so, we have been able to get effectively the sensitivity of the full GLRT with only order $O(n^2 \log(n))$ work. We leave discussion to a future paper.

## 3. APPLICATION: EXTRACTION OF MULTIPLE LINE FRAGMENTS

Suppose we have an image made of many curvilinear fragments and these are embedded in noise. This is more challenging than the beam detection problem because there may be an unspecified number of fragments and because they may be curving. We consider the problem of extracting many beamlets overlapping substantially with the underlying image features.

In order to describe our approach, we need to explain three terms.

*Recursive Dyadic Partitioning (RDP).* Consider an image as a piecewise constant function on a unit square. A recursive dyadic partition is any partition of the square reachable according to the rules (1) that the trivial partition is an RDP (i.e. the partition of the whole square as one piece); and (2) that if we take a partition and subdivide a piece of it into four equal squares, we get a new RDP. Typical examples of RDP's include complete partitions of depth $j$, which have $4^j$ squares of side $2^{-j}$. Other partitions can be extremely spatially homogeneous. A beamlet-decorated RDP is an RDP in which some of the squares of the partition are decorated by single beamlets.

*Models.* Given a BD-RDP, there corresponds a model for noisy data. An undecorated square of the BD-RDP will be modelled as having a zero mean value throughout; a decorated square will be modelled as having a zero value away from the decorating beamlet, and having a possibly nonzero value in pixels intersecting with the beamlet.

*CPRSS.* A popular device for fitting models to data is to optimize the Complexity Penalized Residual Sum of Squares (CPRSS), defined as

$$CPRSS(\lambda, M) = \|y - E(y|M)\|_2^2 + \lambda \#(M),$$

where $\lambda$ is a penalizing parameter, $M$ denotes a model from a pool of candidate models, $y$ is our observed data (image), the notation $E(y|M)$ represents the expectation of $y$ given the model $M$, $\|\cdot\|$ is the $\ell^2$ norm,

$\#(M)$ denotes the complexity of the model $M$. The empirical expectation $E(y|M)$, is obtained by least squares projection on the model subspace.

Some interesting theory proving the near-optimality of CPRSS minimizers is available.[2,1]

There is a fast algorithm to minimize CPRSS in the setting of BD-RDP –the theoretical complexity of minimizing CPRSS is $O(n^2)$. A detailed discussion is available.[1]

In the present paper, we briefly show numerical examples. In Figure 3, we consider the noiseless "Picasso" image, and show results for several different values of the penalty factor $\lambda$, illustrating the varying complexity features extracted. Evidently, the larger the value of the $\lambda$, the coarser the extracted features. This is consistent with our belief that a more severe penalty should lead to a simpler model. We leave to future work a full careful discussion about the connection between the value of parameter $\lambda$ and the surviving features.

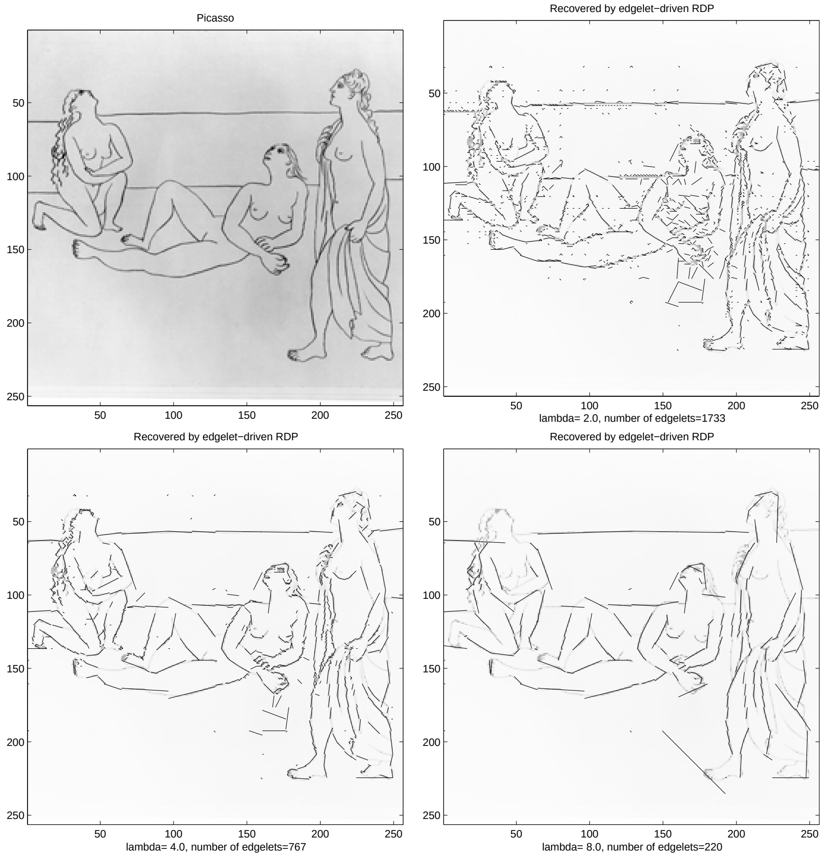

**Figure 3.** Illustration of Beamlet-driven Recursive Dyadic Partitioning (BD-RDP). In a dyadic square, each beamlet survives only after competing with the assumption that there is no edge present. The upper left plot is the original image, and the remaining plots are the BD-RDP with different values of the penalizing parameter $\lambda$, ($\lambda = 2.0, 4.0$ and $8.0$,) by the order: upper right, lower left, and lower right. Correspondingly there are $1733, 767$ and $220$ remaining beamlets.

## 4. APPLICATION: EXTRACTION OF MULTIPLE BLOB FRAGMENTS

The ideas we just sketched for MultiLine extraction also work for MultiBlob extraction. Suppose that we have an image containing indicators of one or several objects embedded in heavy noise. For us, an object is simply a set with sufficiently smooth boundary.

We define the dictionary of wedgelets as follows. On each dyadic square we consider all beamlet decorations, and consider the two fragments of the square obtained by dissection along the beamlet. The indicator of each fragment will be called a wedgelet. There are $O(n^2 \log(n))$ wedgelets generated in total by considering all dyadic squares out to scale $1/n$ using beamlet decorations from $\mathcal{B}_{n,1/n}$.

We now model data associated to a beamlet-decorated dyadic square as a linear combination of the two associated wedgelets. Hence we are saying that on one of the two fragments the data has one mean value, while on the other fragment, the data has another mean value.

A key remark is that we can perform fast CPRSS minimization by exploiting the beamlet pyramid. In effect, for fitting models by least squares, we need to calculate all inner products of the data against all wedgelets. The beamlet transform allows us to calculate all these inner products. Using integration by parts, we consider the beamlet transform of the vector-valued function

$$\mathbf{F}(x,y) = \left[ \int_0^x I(t,y)dt, \int_0^y I(x,t)dt \right].$$

The beamlet pyramid of $\mathbf{F}$ contains within it all the information we need about inner products of wedgelets with $I$. Once we have this information, we can perform CPRSS minimization in this class of models in $O(n^2)$ operations.

In Figure 4, we give an example of wedgelet approximation to noisy data. Note in Figure 4, we can see that the wedgelet approximation closely resembles the noiseless image, even though it is a result of an empirical process only based on the noisy image.

There are a few differences between the approach used for formal theoretical analysis[1] and the one we implemented here. For example, we did not choose the theoretically optimal value for $\lambda$ which had been previously developed.[1] We view our examples as some exploratory experiments. They illustrate substantial promise for automatic blob extraction in very noisy images.

## 5. APPLICATION: AUTOMATIC FILAMENT EXTRACTION

We now consider the problem of detecting a filament in a noisy image. We assume that the filament has certain degree of smoothness and a constant elevation, with the rest of the image at mean zero. We consider the "very-noisy" case where the elevation is low, the noise level large, and the filament itself is not detectible by pixel-level statistics. For an example, see Figure 5. The upper left figure shows the filament image without noise. The upper right image is the noisy picture. Note the curve in the upper right figure is very difficult to discern.

There is an extensive literature on curve finding and following algorithms. Dynamic programming[3](DP) has been used in this connection for decades.

Suppose the two endpoints of the curve are known. We propose to search for a curve such that a *complexity-penalized mean elevation measure* is maximized. That is, we search for a polygonal path connecting the endpoints, along which the elevation is large, and which is, at the same time, not a very complicated path. Associated to each beamlet, we define two terms, an aggregate elevation and a length. The length is the euclidean length of the beamlet. The aggregate elevation is the value of the beamlet transform of the image, penalized by subtraction of a term which is the penalty factor times the square root of the length of the beamlet. The penalty factor is determined by the image size and noise level. The CPEM-optimal curve is that chain of beamlets starting at one endpoint and ending at the other which maximizes the cumulative ratio (aggregate elevation)/(aggregate length) Our proposed criterion has a theoretical optimality which we are unable to discuss here.

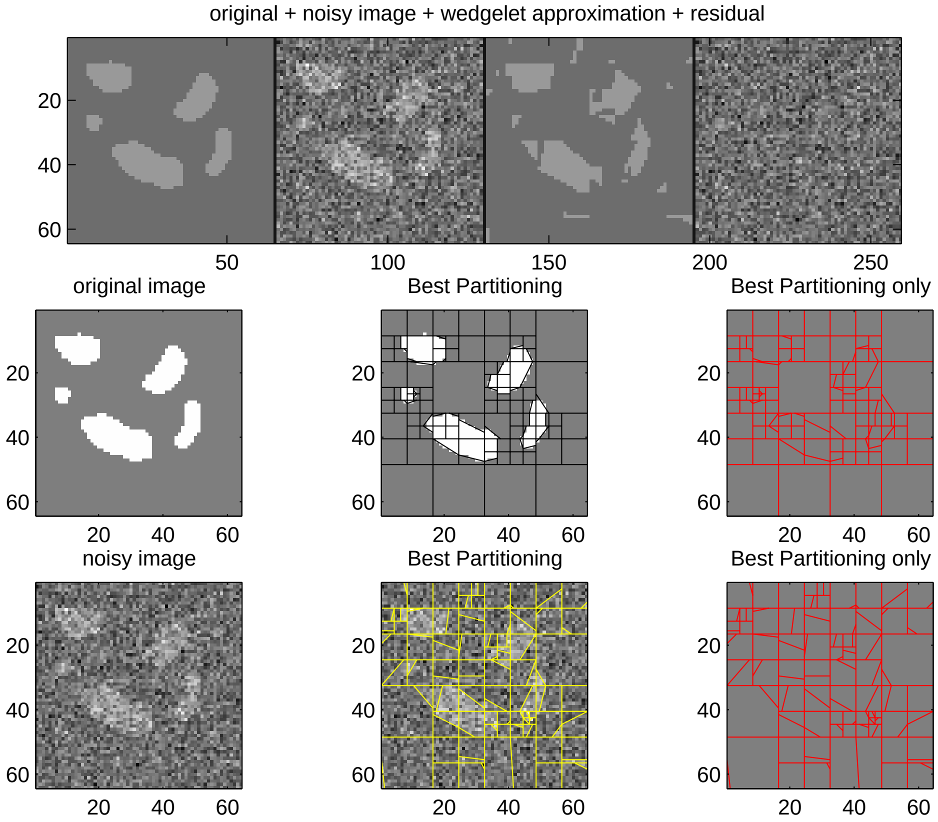

**Figure 4.** Multiple Object Detection by using Wedgelet Approximation. The wedgelet approximation is based on a beamlet-decorated recursive dyadic partition (BD-RDP). The panel on the first row shows, from left to right, the original image, a noisy image by at each pixel adding a Gaussian noise with standard deviation $\sigma = 2/3$, the wedgelet approximation to the noisy image, and the difference between the noisy image and the approximation. On the second row, the first plot (from left) is the original image, and the other two give optimizing BD-RDP. The last row is for the noisy image.

This optimization problem is a network flow problem,[4] of finding the Minimum Cost To Time Ratio Path (MCTTRP) in an undirected, doubly weighted network. Here the network is the beamlet graph, the cost is the aggregate elevation and the time is the length.

In general, such MCTTRP problems are NP hard, with no guarantee of polynomial time solutions. We developed a heuristic method to solve our problem inspired by the shortest path algorithm. We exploited a special fact about the beamlet pyramid – sparse representation of paths by chains of a few beamlets. In traditional applications of network flow to image analysis, the underlying network connectivity involves only near neighbors, so a beam of global scale requires $O(n)$ links. In the new approach, a beam of global scale requires only $O(\log(n))$ links. There can be very few steps of 'link propagation' before termination.

We give an example showing that in a case where traditional network flow algorithms based on the minimum distance path approach do not work, the CPEM approach can still find the embedded curve. In figure 5, the lower right subfigure is the estimation from the minimum distance path, the lower left one is the estimation from the CPEM approach.

We stress a certain feature of these figures: the multiscale nature of the optimal path, which uses a few long straight links in areas of the curve which are long and straight and which uses many short links in areas of rapid turning. The algorithm has automatically chained together a multiscale representation of

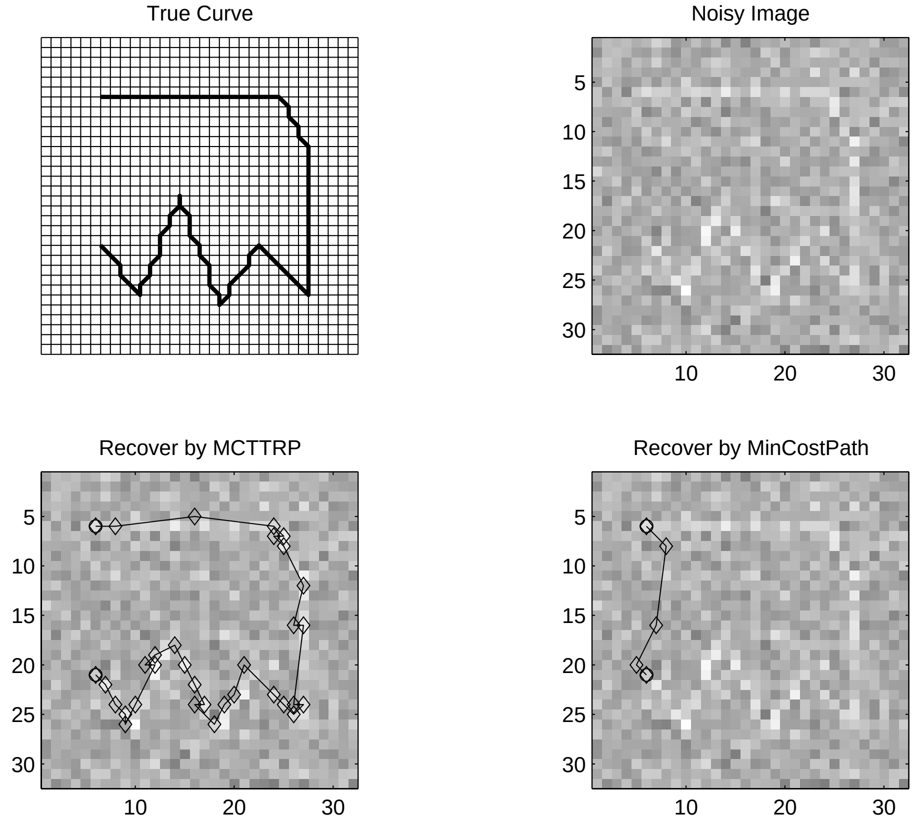

**Figure 5.** A case when the minimum cost path fails but the MCTTRP succeeds. The magnitude of the original curve (shown in upper left figure) is 1. The standard deviation of the additive Gaussian noise at each pixel is 1/2. The result of the MCTTRP method closely recovered the underlying curve (lower left) while the minimum cost path approach could not (lower right).

the curve of interest.

## 6. APPLICATION: AUTOMATIC OBJECT EXTRACTION

We can use beamlets to extract objects – not just filaments. At first glance this seems unlikely, since a curve is one dimensional, and so tangibly related to beamlets, while an object is, two dimensional, and so not obviously related. However, as in the case of wedgelet modelling, by integration-by-parts, one can transform regions into their boundaries, and so convert one problem into the other.

Suppose our object is the indicator of a region in the image with a positive constant elevation and smooth boundary. Conceptually, we would like to maximize the following ratio:

$$\{\text{integral over } \mathbf{R}\}/\{\text{the length of the boundary of } \mathbf{R}\}. \tag{1}$$

This can be motivated in two ways. First, thinking of matched filtering and the Generalized Likelihood Ratio Test (GLRT): if we were searching among a limited number of possible regions, we would prefer the region $\mathbf{R}^*$ maximizing the following ratio:

$$\{\text{the integral over } \mathbf{R}\}/\{\sqrt{\text{the area of } \mathbf{R}}\}.$$

Now when the shape of the region is given, the length of its boundary is proportional to the square root of its area; so there is a crude relationship to GLRT/Matched Filtering ideas. Second, the important issue will of course be determination of the boundary shape, noise will obviously affect this by making the boundary wiggly; penalization of excessive boundary length explicitly regularizes the shape of the recovered boundary.

As it turns out, when we pose the problem in the above form, it can be solved by linear programming methods, which makes available to us a variety of fast interior point methods. The connection with linear programming goes in two steps. First, using integration-by-parts, we can transform it from a problem about areas over regions into a problem about integrals along polygons. The resulting problem, as in the previous section is of Minimum Cost to Time Ratio type. However, because the boundary of a region is a cycle, the sought-for object is a directed cycle rather than an undirected path between specified endpoints. This change makes a profound algorithmic difference. From the theory of network flows, we know that the Minimum Cost-to-Time Ratio cycle problem is amenable to solution as linear programs. The solution of MCTTRC by linear programs goes back to Dantzig-Blattner-Rao in the mid 1960's. We omit further discussion here.

We instead briefly explain the integration-by-parts transformation. The Gauss-Green theorem says that for a region $\mathbf{S}$ with boundary $\partial\mathbf{S}$, we have

$$\int_{\mathbf{S}} \nabla \cdot \mathbf{F} dA = \int_{\partial\mathbf{S}} \mathbf{F} \cdot \mathbf{n} ds,$$

where $\mathbf{F}$ is a vector field, $\nabla \cdot \mathbf{F}$ is its divergence, $A$ is the area variable, $\mathbf{n}$ is the normal direction vector, and $s$ is the parameter of the boundary. Choose the vector field $\mathbf{F}$so that its divergence is equal to the negative of the image – $\nabla \cdot \mathbf{F} = -I$, where I  denotes the image – then the integration on the left of the Gauss-Green identity is simply $\int_{\mathbf{S}} -I \; dA$, from which we can obtain the numerator in (1). The vector field

$$\mathbf{F}(x,y) = \left[ \frac{1}{2}\int_0^x I(t,y)dt, \frac{1}{2}\int_0^y I(x,t)dt \right]$$

satisfies the divergence condition.

Now suppose in our treatment of discrete data we only consider regions with boundaries that are cycles of beamlets. On each beamlet, the normal direction $\mathbf{n}$ will be constant. So computing the value of $\mathbf{F} \cdot \mathbf{n}$ can be done through the beamlet transform of each coordinate of vector field $\mathbf{F}$.

Now call the integration of $\mathbf{F} \cdot \mathbf{n}$ over a beamlet the "cost" of the beamlet, and the length of the beamlet the "time" of the beamlet. The solution to the problem in (1) is actually the MCTTRC in the beamlet graph.

The theoretically advantage of this approach is that when the object has a regular boundary, there will be rather good approximation to its boundary by a chain of relatively few beamlets, and there will be excellent signal-to-noise ratios of the underlying data, leading to high accuracy reconstructions.

In Figure 6, we give a numerical example. On the first row, the left plot is the noiseless object, the middle plot is an image contaminated by noises, the right plot shows the estimation via MCTTRC. The second row is the simulation result at a different noise level. Note in the first case, MCTTRC almost find the object perfectly, although the object is fainted in the noisy image. In the second case, there is a distortion on the left part of the estimation. This is due to fact that the noise level is too high. A careful study of the noise level at which the MCTTRC is still capable of recovering the embedded object will be a future research topic.

We note that the use of integration-by-parts to transform a problem of objects to a problem of polygonal cycles was previously developed by Ishikawa and Jermyn.[5] Our work differs due to the multiscale nature of the beamlet graph.

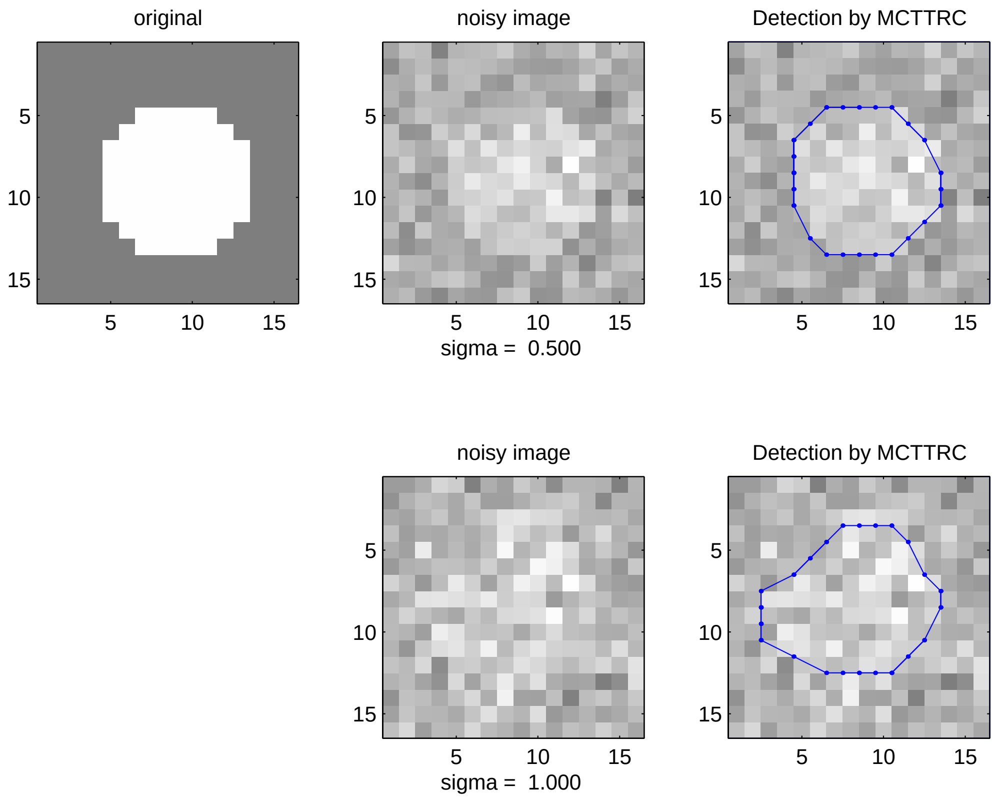

**Figure 6.** MCTTRC to detect an object in a very noisy image. We use the idea of maximizing the ratio between the integration of a 2-D function within a region and the length of its boundary. By Gauss-Green and network flow theorems, this can be formulated as a Minimum Cost-to-time Ratio Cycle (MCTTRC) problem. The first plot on the top row is the original image. The noisy images are made by pixelwisely adding Gaussian random variables with standard deviation $\sigma$. The cycles in the images in the third column are the results of our MCTTRC. MCTTRC is solved by linear programming.

## 7. CONCLUSION

Beamlets introduce a new multiscale organizing principle in image analysis, with some important theoretical benefits, not reported here, and a variety of applications, a few of which we have sketched. Our algorithms automatically chain together beamlets to create a polygonal path in the image, using multiscale combinations of short and long beams to best track filaments, edges, etc. Hence, we have come across a constructive new approach to the problem of adaptive multiscale curve following, and a theoretical edifice which explains the near-optimal performance of the approach at poor S/N. Numerical simulations seem to show that beamlets are not merely of theoretical interest – they truly can detect and extract lines, curves, and blobs at previously forbidding SNR.

## ACKNOWLEDGMENTS

This research was partially supported by National Science Foundation grants DMS 98–72890 (KDI); by AFOSR MURI-95-P49620-96-1-0028, and by DARPA BAA-98-04.

## REFERENCES

1. D.L. Donoho, “Wedgelets: Nearly minimax estimation of edges,” *Annals of Statistics*, vol. 27, no. 3, pp. 859–897, 1999.

2. D.L. Donoho, "Cart and best ortho basis: A connection," *Annals of Statistics*, vol. 25, no. 3, pp. 1870–1911, 1997.
3. 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*, vol. 17, no. 3, pp. 294–302, 1995.
4. Ravindra K. Ahuja, Thomas L. Magnanti, and James B Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993.
5. Ian Jermyn and Hiroshi Ishikawa, "Globally optimal regions and boundaries," in *7th ICCV*, Kerkyra, Greece, September 1999.
6. Richard M. Karp, "A characterization of the minimum cycle mean in a digraph," *Discrete Mathematics*, vol. 23, pp. 309–311, 1978.
7. R.M. Karp and J.B. Orlin, "Parametric shortest path algorithms with an application to cyclic staffing," *Discrete Applied Mathematics*, vol. 3, pp. 37–45, 1981.