Computer Science
Technical Report

Automated Velocity Picking: A Computer Vision and Optimization Approach.

Darrell Whitley, Ross Beveridge, Charles Ross

November 1998

Technical Report CS-98-114

Computer Science Department
Colorado State University
Fort Collins, CO 80523-1873

Phone: (970) 491-5792 Fax: (970) 491-2466
WWW: http://www.cs.colostate.edu

FINAL REPORT

Colorado Advanced Software Institute

A COMBINATORIAL OPTIMIZATION ALGORITHM FOR SEMBLANCE VELOCITY PICKING

Principal Investigators:	Darrell Whitley and J. Ross Beveridge Computer Science Department Colorado State University Fort Collins, CO
Graduate Student:	Charles Ross Computer Science Department Colorado State University Fort Collins, CO
Collaborating Company:	Landmark Graphics Barry Fish Company Representative

Contents

Automated Velocity Picking: A Computer Vision and Optimization Approach.

Darrell Whitley, Ross Beveridge, Charles Ross

November 1998

Abstract

"Velocity Picking" is the problem of picking velocity-time pairs based on a coherence metric between multiple seismic signals. Coherence as a function of velocity and time can be expressed as a 2-D color image representing the "Semblance Velocity." Currently, humans pick velocities by looking at the Semblance Velocity image; picking velocities for a seismic survey can take days or even weeks. Automating the process as pure optimization without exploiting the Semblance Velocity image yields an essentially intractable problem. The problem can also be posed as a geometric feature matching problem similar to those used in computer vision. A feature extraction algorithm can recognize islands (peaks) of maximal power corresponding to velocities in the Semblance Velocity image: a heuristic combinatorial matching process can then be used to find a subset of peaks which maximizes the coherence metric.

Our results indicate this combinatorial approach has many advantages. It is fast, in as much as the evaluation process is restricted to a small finite set of line segments connecting peaks in the image. It also allows the peak selection process to be interactive. Users can hand select peaks; the search then is restricted to solutions consistent with the peaks selected by the user. Our experience indicates that selecting even a single peak is enough to restrict the search to guide it to very good solutions.

We also introduce another way to differentiate competing solutions. We compute an initial set of solutions, then compute a composite median solution across the set. Because geology is such that we generally expect gradual change in rock strata over short distances in space, solutions far away from the median are likely to be incorrect. After obtaining the median, we do a second pass of optimization in which "closeness to the median" is included as an additional optimization criterium. The final results are similar to those produced by humans and in fact produce a higher evaluation than human picks in terms of the resulting coherence across the seismic signals.

1 Preface

The following report summarizes our results from September 1, 1997 to September 1, 1998. The original project was approved as a 2 year project. However, the CASI rules concerning the amount of money that companys contribute to a project changed in mid-project. This caused the company's dollar cost in terms of matching rate to jump from 10 percent in year 1 to 45 percent in year 2. The also happened at such a time that Landmark had not budgeted the additional funds needed to continue the project at the higher matching rate. The project was approved for continued funding by CASI, but the project was nevertheless terminated after one year due to lack of matching funds.

We still was able to complete the prototyping of the system which we had proposed. We in fact did much more than we had scheduled for year 1 in an effort to get as much done as possible to show the merit of our approach. The project is also remarkable in as much as the basic methods outlined in the original CASI proposal worked extremely well–so that the work reported here very closely follows the methods in the original proposal.

Because our evaluation of this systems was largely scheduled to be done in year two, we have combined the Results and Evaluation section.

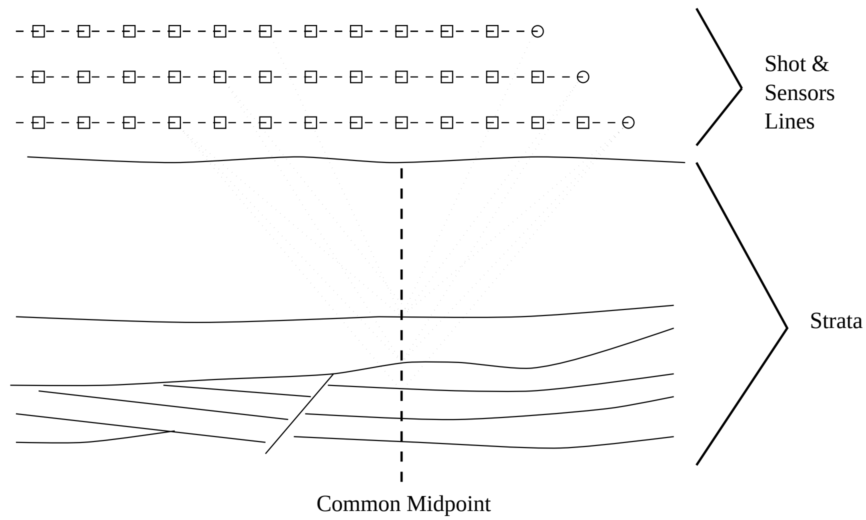


Figure 1: *Example of a common midpoint gather.*

2 Objectives

During a seismic survey, a source (e.g. a dynamite shot) and multiple receivers (e.g. geophones) are used to record seismic signals. Data from many sources and receivers arrayed over the survey area provide indirect information regarding each "reflecting layer" of earth strata. To reconstruct the underlying strata, geophysicists must correct for the distance between different sources and receivers and combine data from multiple signals into a **common midpoint gather**. In effect, the common midpoint gather is a restructured signal that models what a seismic signal would look like if it had been collected by a source and receiver both sitting at exactly the same spot (e.g., the common midpoint in Figure 1) on the earth's surface.

To assemble a common midpoint gather, the average signal velocity from the earth's surface down to a specific reflecting layer must be estimated. Guidance for picking velocities is obtained from a 2-D Semblance Velocity image which encodes the "power", or cross-correlation, between all signals involved in the common midpoint gather. For our purposes here, time encodes the depth to a particular reflecting layer. The greater the power for a velocity-time pair, the more coherent are the signals in the seismic survey. The image on the left of Figure 2a shows is a Semblance Velocity image in which hotter colors (e.g. red) indicate higher power.

While the "islands" representing peak power in the Semblance Velocity image suggest good velocity estimates, simply chaining together all plausible peaks is inadequate. There are many complicating factors, including echos and artifacts produced from complex geophysical structure. Also, velocities usually increase with depth. Geophysicists use these and other principles to guide their choice of peaks: an example is illustrated by the red line in Figure 2b. Based upon this selection, the seismic signals shown on the right in Figure 2 are adjusted. The selected peaks must satisfy certain structural properties in the Velocity image. They must also yield a high quality common midpoint gather, which is related to the quality of the adjusted signals shown on the right side of Figure 2. If all plausible peaks can be extracted automatically, and the principles used by the geophysicists quantified, then a heuristic combinatorial optimization algorithm should be able to select the "best" subset of peaks.

"Velocity picking" is largely considered to be an unsolved problem, with the state-of-the-art involving humans picking velocities based on the Semblance Velocity image and supplemental graphics (the right hand sides of Figures 2a and 2b) showing how selected velocities impact the collection of associated seismic signals. Picking velocities by hand for an entire survey can take days or even weeks. Landmark has developed neural networks for automated velocity picking [6], but their approach requires hand processing to generate the training data and the results are less than adequate. At the time that this project began, Landmark considered velocity picking to be one of the most important remaining large problems that does not have a reasonably good automated solution. Our objective was to develop an automated solution for this problem.

We have since learn that at least one large oil company has been using splines to fit a surface through the set of Semblance Velocity images representing a survey. This approach does not pick individual velocities, but rather creates a line through each Semblance Velocity image. By treating the set of images as a volume, consistency from one Semblance Velocity image to the next is insured. Because this method is proprietary, we have no software for this method. (Landmark also currently does not possess this technology.) We thus have no comparative data with the methods developed here. However, we can outline the likely advantages of the approach taken in our research. These are discussed in the final section of this report.

In our approach, a set of initial candidate peaks are found in the Semblance Velocity image, and then a heuristic optimization procedure will select the best subset of candidate peaks. Identifying candidate velocity peaks in the Semblance Velocity image is an example of an image feature extraction task similar to peak detection as commonly performed in thermal imagery for Automatic Target Recognition [5] and closely related to model-driven feature detection [9, 7, 10].

By design, feature extraction selects more peaks than are strictly necessary; subsequent optimization can remove false positives but cannot recover if a true peak is missed. An objective function quantifying constraints associated with both the structure of the peaks in the Semblance Velocity image and the quality of the resulting common midpoint gather codifies the "best" set of peaks to pick.

2.1 Background

"Two-way zero offset time" is the amount of time required for a signal to travel from a source down to a reflecting layer (e.g. a transition between rock layers) and back up to a receiver, assuming there is zero offset between the source and receiver. In other words, the source and receiver are both located at exactly the same position. In practice, receivers do not have zero offset from sources (for example, the source might be a shot of dynamite), and instead multiple receivers are placed at various distances from the source; receivers may be kilometers from a source.

When there are non-zero offsets, this results in a travel time curve as a function of the offset. Under the simplifying assumption that velocity is constant through the earth, the travel time curve will be hyperbolic and all signals may be adjusted and combined to form a common midpoint

(CMP) gather as shown in Figure 1. John Scales gives an excellent overview of these concepts [13].

To construct a common midpoint gather, we must calculate a reasonable approximation for the two-way zero offset time. This is done in the preliminary stages of data processing, before more detailed and complex corrections are made. For example, "static corrections" associated with near surface variation are performed after initial construction of common midpoint gathers.

Figure 1 illustrates the right triangle formed by the source (or receiver), the point of reflection and the point directly above the point of reflection. We know the offset distance and the measured travel time from source to receiver. Assuming there is a known constant transmission velocity for the signal passing through the material above the reflecting layer, the Pythagorean theorem lets us calculate the distance between the reflection point and the point directly overhead on the earth's surface and thereby to infer the two-way zero offset time of the common midpoint gather.

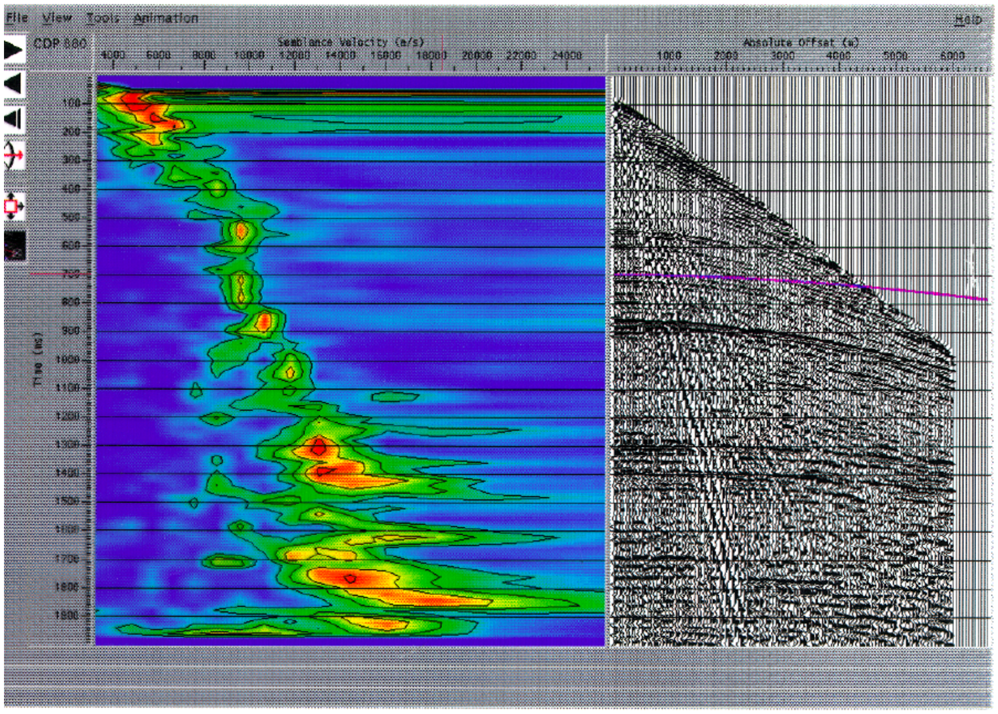
In reality, the velocity directly above the reflecting layer is neither known nor constant. However, since many sources and receivers are present, the highly redundant data suggests that we can look at the coherency between sets of signals based upon different hypothesized piece-wise constant velocities. We can in fact look at coherency over all possible velocity-time pairs as measured by signal power (i.e. the cross-correlation between the signals). A discrete sampling of power measured over the 2-D velocity-time space yields the Semblance Velocity image illustrated in Figure 2. The x-axis is velocity and the y-axis is time. For our purposes, time can be thought of as distance or depth. The power associated with a particular velocity-time pair is represented by color: blue-green-yellow-red shifts correspond to a increasing power (blue = low, red = high). The red and yellow "islands" in Figure 2 are thus the most likely velocity-time pairs corresponding to the geology. We can also calculate overall post coherency after velocity picking.

2.2 The Velocity Picking Problem

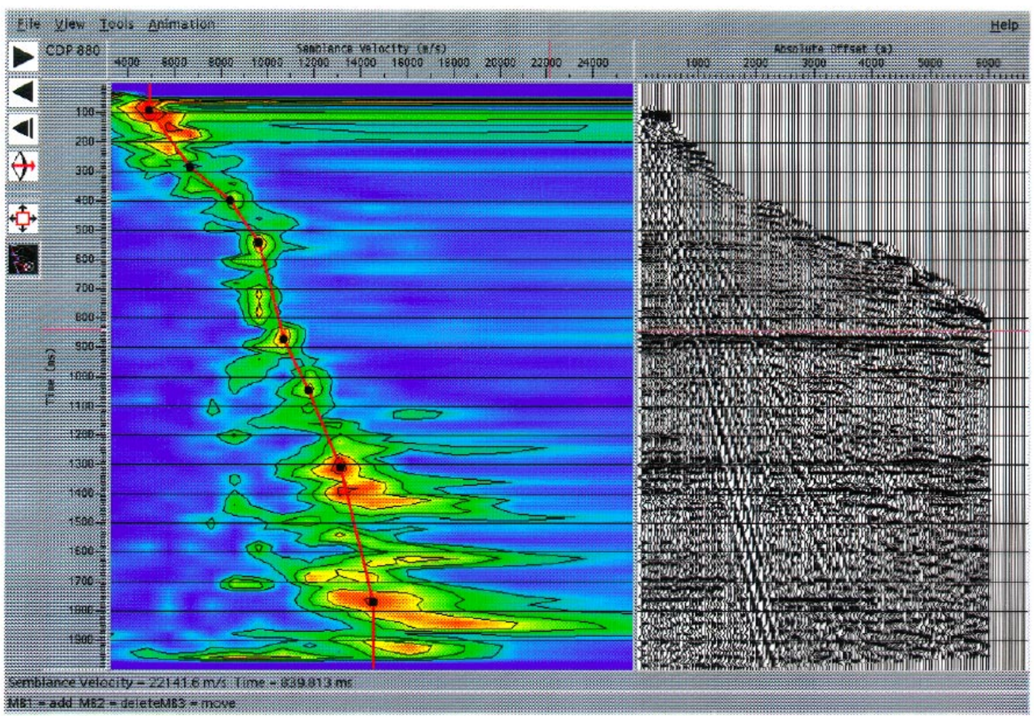
The velocity picking problem is to pick the velocity-time pairs that maximize power while also accounting for the observed data. In the bottom image in Figure 2 the dots correspond to hand selected velocity peaks. The black and white images on the right correspond to the actual signals. One can see in the top image the curve in the signals and one can see in the lower image that the signals have been "flattened." This corresponds to the adjustment associated with the calculated two-way zero offset times derived from the selected velocities.

In Figure 2, not all "islands" of peak power are selected. While the most appropriate velocities are those that maximize power, there are other factors that must be taken into consideration. For example, there can be reflections which, for the purposes of this proposal, can be thought of as echos. Islands of increased power that occur at the same velocity but later in time are most likely echos of the original pulse that have bounced off of another reflector. Additional artifacts can be introduced into the Semblance Velocity image from a variety of unmodeled signal sources.

Based upon our prior experience with computer vision problems, our approach to the velocity picking problem is to divide the task into two parts. First, a feature extraction algorithm identifies a superset of possible peaks in the Semblance Velocity image. Subsequently, a combinatorial optimization process searches the powerset of possible peaks to find a subset which is 'best', where best is defined relative to a criterion function. This criterion function will combine preferences in



(a)



(b)

Figure 2: *Example of Semblance Velocity.*

terms of the form the velocity curve (see the red line in Figure 2b for an example) and the quality of the resulting coherence of the adjusted seismic signals, as measured by the energy under the line which results from connecting the selected peaks in the Semblance Velocity image.

Candidate Velocity Peak Extraction. We use computer vision methods to pick the "islands" in the Velocity Semblance image. We wish to minimize the potential for false negatives. A false positive can be later removed during a peak selection phase, but a peak missed at this stage cannot easily be added back into the process.

This work leverages our experience based closely on two methods developed for similar tasks. The first is analogous work in peak detection for Automatic Target Recognition in thermal imagery. This is a well studied task [5]. The second relates to a broader class of model-based feature detection/extraction algorithms [12, 9, 7, 10]. Both of these approaches combine geometric and statistical constraints defined over image areas to 'model' the event of interest and thereby detect it: in our case these are the peaks or "islands".

Finding the 'Best' Subset of Candidate Peaks. The task of finding an optimal, or best, subset of peaks is similar to many of the geometric matching problems with which we have considerable experience. Our background work has looked at optimal matching of 2D line segment models to cluttered and complex line data [4, 3], matching 3D line models to 2D image features assuming 3D perspective projection [1, 2, 8], optimal matching of 3D models to multi-modal data [11], and recent advances in combinatorial line matching using local search within genetic algorithms [14].

To clarify our problem formulation, let $\mathcal{Q}$ be the set of n peaks extracted from the Semblance Velocity image. The best solution to the velocity picking problem is a subset s^* of peaks that maximizes some objective function F.

$$F(s^*) \geq F(s) \quad \forall s \in 2^{\mathcal{Q}} \tag{1}$$

In other words, the best velocity pick is the set s^* from the 2^n elements of the powerset of $\mathcal{Q}$.

The objective function F must blend soft constraints from two information modalities: energy information from the Semblance Velocity image (E) and constraints (C) on the way in which "peaks" in the semblance velocity image can be connected.

$$F(s) = F_{\mathrm{E}}(s) + F_{\mathrm{C}}(s) \tag{2}$$

This objective function, F_{E}, is an existing function used by geophysicists; it is the amount of energy under the line connecting "peaks" in the Semblance Velocity image. Maximizing energy under the line in effect maximizes the coherence of the signals involved in the common midpoint gather after adjusting for velocity. However, our research shows that some solutions that maximizing this energy function are also non-feasible solutions. These solutions represent unrealistic geological interpretations of the data.

Constraints are added to the objective function to match expectations about geological feasibility. For example, velocity tends to increase over time, the rate of change increases rapidly at shallow depth and velocity changes are generally smaller at greater depth. Finally, velocity changes are relatively smooth and velocity changes are generally consistent from one semblance velocity image to the next in a seismic survey. All of these expectations are encoded as constraints incorporated into the objective function.

3 Approach: Exploiting The Semblance Velocity Image

A semblance velocity image is an n by m image in which the y axis represents time (time is read from top to bottom of the image), and the x axis represents velocity (with increasing velocities to the right). Each value in the image represents the coherence of the seismic signals at that point in time when the signals are adjusted using the corresponding velocity value. The resulting value is in the range $[0, 1]$. Values near 0 mean there is little or no evidence of waves traveling at this particular velocity, while values near 1 represent significant evidence. An example semblance velocity image is shown in figure 3. Note that the "peaks" in this image are much less pronounced and distinct than those in Figure 2. The data sets given to us by Landmark (as illustrated by in figure 3) in fact represent very difficult velocity picking problems.

After velocities are picked, a polyline is drawn through the image. This line represents the changes in velocities at different depths in the seismic survey. Figure 4 shows a semblance velocity image with the corresponding line superimposed on it. The points that lie on the line represent the selected velocity at each depth.

Currently, the semblance velocity picking is often done by hand. Experts familiar with the local geology and the behavior of geophysical systems sit in front of a computer terminal to pick velocities. This is a very demanding and time consuming process. The expert performing the calculation begins by viewing a semblance velocity image that has been enlarged through linear interpolation. The image is usually artificially colored in order to enhance contrast and improve viewing. The person then selects points which represent geologically plausible key points in the image. As each point is selected, the computer displays what effect the corresponds set of velocities have on the seismic signals (again see the leftmost set of seismic signals in figure 1.) Using this feedback from the computer, the operator further refines these selections and eventually produces a line connecting velocity picks which are supported by both the data and knowledge about the geology.

The expert then proceeds to the next image and the process is repeated. The seismic imaging process characteristically produces large amounts of data, and there are often thousands of images that must be analyzed in this painstaking way. Each image takes about a minute of expert operator time to process. Because geology does not change rapidly over a short distance and because semblance velocity images can be sorted so as to be spatically adjacent, velocity picks tend to be similar from one image to the next, as well as similar over a single seismic survey.

3.1 Problem Definition

The goal is to take any semblance velocity image as input and produce a set of velocity picks similar to those produced by an expert. An automated system has the disadvantage in that it must

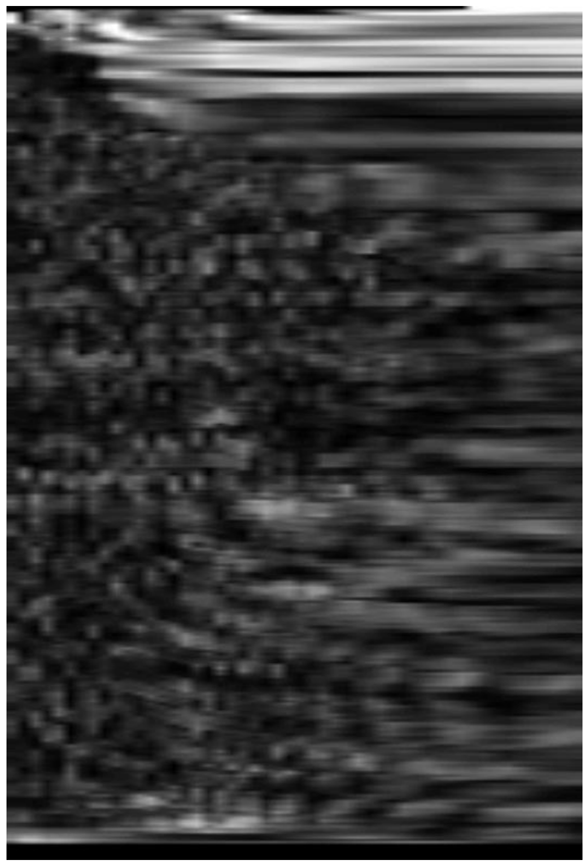


Figure 3: Sample Semblance Velocity Image

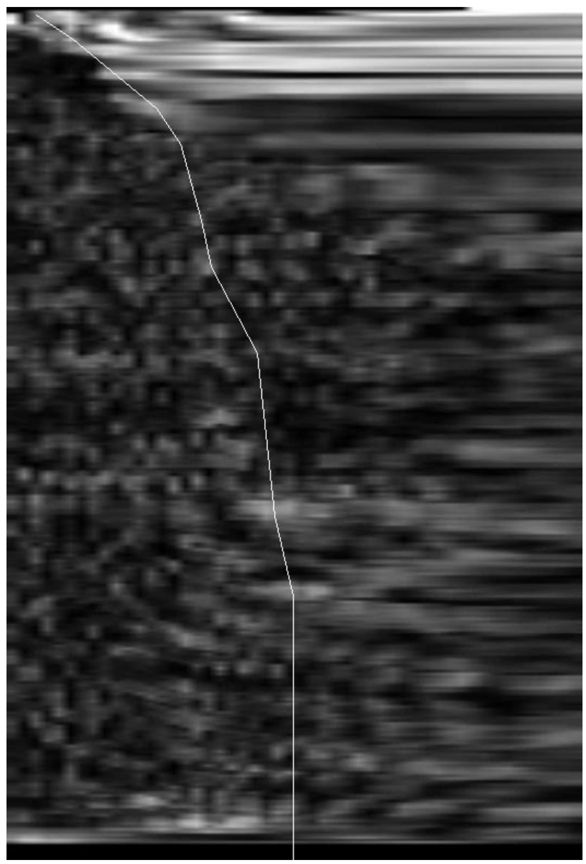


Figure 4: Sample Semblance Velocity Image with a polyline connecting peaks in the image.

solve the problem without utilizing the local geological knowledge that a human operator would use. An example of some of this missing data might be strong geological expectations based on the type of site being surveyed. For example, aquatic surveys and land surveys have very different characteristics, and the human can take advantage of these differences to include a bias as a result of a predetermined expectation. Surveys in the Rocky Mountain region, for example, are also likely to display more variation in geology within a seismic survey.

3.2 Pre-processing Phase

There are three pre-processing steps. The information gathered in these three stages is subsequently used during the velocity picking phase. Pre-computing this data dramatically improves the runtime speed of the program. The three steps of pre-processing are: Peak Detection, Length Computation, and Energy Computation.

3.3 Peak Detection

The first, and most important step in pre-processing is Peak Detection. During this phase it is best to think of the semblance velocity image as a surface. The value at each pixel represents the elevation at that particular point on the surface. Bright areas of the image correspond to mounds or hills on a landscape surface.

Every pixel in the image has a value representing its height in the elevation map. Our goal is to find peaks. In this context a peak is any local maxima, or any pixel whose height is greater than its surrounding eight neighbors. Pixels who have an equal elevation will be handled separately, so for now we can ignore these cases.

From any pixel which is not a peak there must exist a path which leads to a peak along which the height monotonically increases. In other words, from a non-peak, it is an uphill climb all the way to a peak. By a path, we mean a sequence of pixels each of which is adjacent to the next. A neighborhood of size eight is used: North, South, East, West, NorthEast, NorthWest, SouthEast and SouthWest.

Imagine every pixel sending out a "scout" who continues to climb uphill until it reaches a peak. The number of "scouts" who arrive on any given peak is an indication of the size of the basin of attraction of the peak. If every pixel voted for the peak to which it was linked by this path, the number of votes would correspond to the size of the peak. The goal of our peak detection algorithm is to both find all of the peaks and to measure their size by accumulating votes. While this could be done by a direct implementation of the algorithm as described, to do so would be inefficient. The actual algorithm used proceeds as follows.

During the peak detection phase, each pixel must store 3 values. The three values are denoted c, d, and v. The value c represents a count of the number of votes that each pixel has currently. d is a vector used to point to another pixel in the velocity semblance image. v always stores the elevation at the point in the image pointed to by d.

To begin Peak Detection, every pixel is initialized so that $c = 1$, and d is pointing to itself.

This initial state treats each pixel as if it were a peak. After each pixel is initialized, two passes are repeated on the entire image. These two passes are then repeated as necessary.

During Pass 1, each pixel whose d value points to itself looks at the surrounding 9 pixels (including itself), and sets its d to point to the one with the greatest v. In the case of a tie, the first one encountered is used. (Each pixel searches itself first, then proceeds to the rest.)

In Pass 2 the d values are allowed to "flow" to the highest point, and votes are cast. This is accomplished by following the d values until a d is found that points to itself. The ending point is then stored back into the original pixel's d value. Each pixel then adds its c value to the c of the pixel pointed to by d, and sets its own c to zero.

These two passes are then repeated until none of the pixels d or c values change. Figure 5 shows an example of this algorithm being performed on a small image region. Note that after the first round, the lower left pixel has not changed. This is because it has no neighbors who have a higher elevation. However, after the second round, it has chosen to move right, instead of up. This is because the pixel to the right will eventually lead to a peak with $v = 9$, while the two pixels above, lead to a peak with $v = 8$. A similar situation happens in the upper right corner of the image.

At the conclusion of this process, each pixel which has a $c > 0$ searches for other pixels with a positive c value in the surrounding area of equal elevation. Any pixels found are merged by summing their c values, and placing the sum into a pixel which is the average of the contributors. Each of the contributors c values are set to 0. This properly handles plateaus, or large areas where the elevation is constant.

At this point, any pixel with a positive c, is called a "peak." Each is then entered into a list of peaks. Each peak carries its c and v values with it, while the d is discarded, since each peaks d points to itself.

Generally, more peaks are found than are necessary–especially very small peaks that are not signicant features in the image. To reduce the number of peaks to a manageable number the peaks are sorted according to the product of their c and v values. Again, c is the count of the votes, and v is the height of the peak. By taking the product of these two values, we can emphasize the need for a peak to have both a large basin of attraction as well as large magnitude. The 200 peaks with the highest product are kept and the rest are discarded. The peaks are finally sorted into y coordinate order, sub-sorted by x coordinate in the event that multiple peaks exist on the same row.

Using this method of Peak Detection, the computer can find single points that represent the areas with high potential of being true geological features. A semblance velocity image with computer-detected peaks superimposed on the image is given in figure 6. (Color examples are also given in figures 8 and 9.) The circles in this image represent the relative size of the basins of attraction, while the gray scale represents the magnitude (the lightest areas have highest magnitude).

3.4 Lengths

Following the peak detection, the next step is the calculation of the distance between every useful pair of peaks. A pair of peaks is defined as "useful" if the second peak lies below and to the right

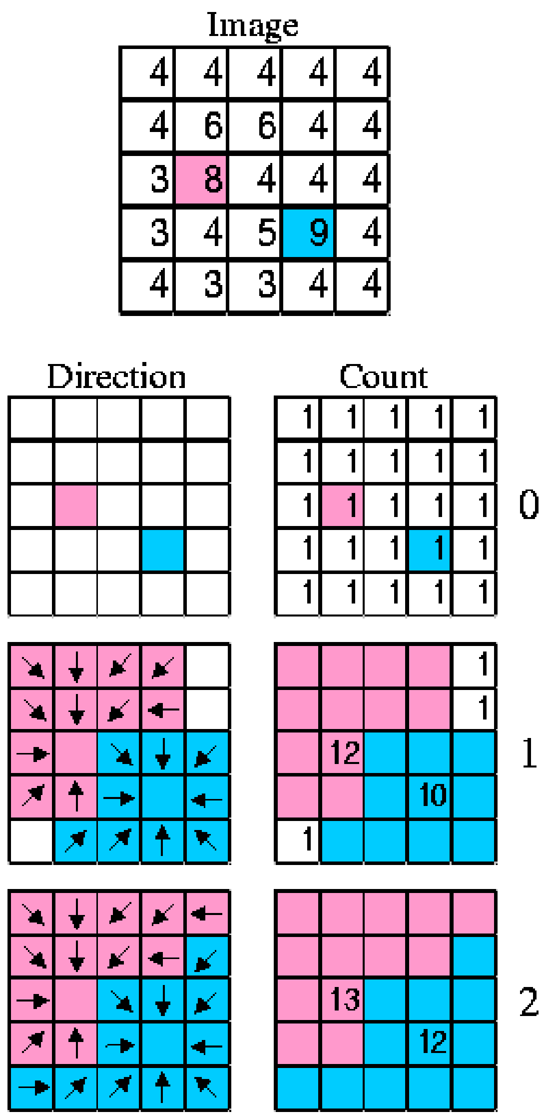


Figure 5: Peak Detection Algorithm

Figure 6: Sample Image with Peaks

of the first peak. We will treat the peaks in directed fashion so that we sort the peaks by depth. A peak is connected to a second peak at greater depth than (i.e. below) the first. We also do not allow the line segments connecting peaks to travel to the left. Travel to the left would mean that the sound waves are slowing down as they go deeper into the earth—a phenomenon which is very rare. Such effects can generally only be predicted and modeled by an expert with specific knowledge of the local geology. In effect, this allows us to consider only those line segments connecting peaks such that the line segments have a non-negative slope. For all pairs of peaks connected by line segments with an non-negative slope, the distance between them is computed and stored. For the infeasible pairs of peaks, a length of 0 is stored as a place holder.

3.5 Energy

More energy under the the line is better than a similar solution with less energy under the line. The energy under the line corresponds to greater coherence in the corresponding set of adjusted seismic signals. This means that the line should pass over more bright pixels than dark ones. This is accomplished by computing the energy under each feasible segment multiplied by its length. Every pixel which lies within distance 2 of the segment is averaged into the total, with its weight computed by $1 - \frac{distance}{2}$. This average is then multiplied by the length of the segment and stored for later use.

3.6 Execution and Interface Issues

At the time of execution the user interacts with our velocity picking system via a Graphical User Interface (GUI), as shown in Figure 7. Within the context of the GUI, the user can view each of the images, perform a variety of automated searches on either a single image or an entire image set, and aid in the search if it becomes necessary.

If the user desires, picks can be selected manually, or any amount of computer assistance can be used to refine the search. The user also has the flexibility to select or de-select peaks, as well as to force a particular peak to be used or not be used during the computer search. All of this is done by directly clicking on the circled area around a peak in the semblance velocity image after it has been loaded into the GUI.

3.7 Search

During the search for a best subset of peaks, a solution is encoded as a string whose length is equal to the number of peaks. Each cell in the string can take on one of 4 values: ON, OFF, $ALWAYS$, or $NEVER$. The ON and OFF setting means that the search may add a peak (ON) or remove a peak (OFF) from the polyline that makes up the connected set of peaks. $ALWAYS$ means that the user has indicated that a peak must be included in the solution and $NEVER$ means the user has in effect removed a peak from consideration; the peaks marked as $ALWAYS$ or $NEVER$ are not manipulated during search. Since the peaks are sorted first by their y coordinate and then sub-sorted by x, it is impossible to encode an unfeasible solution and by checking when a peak is

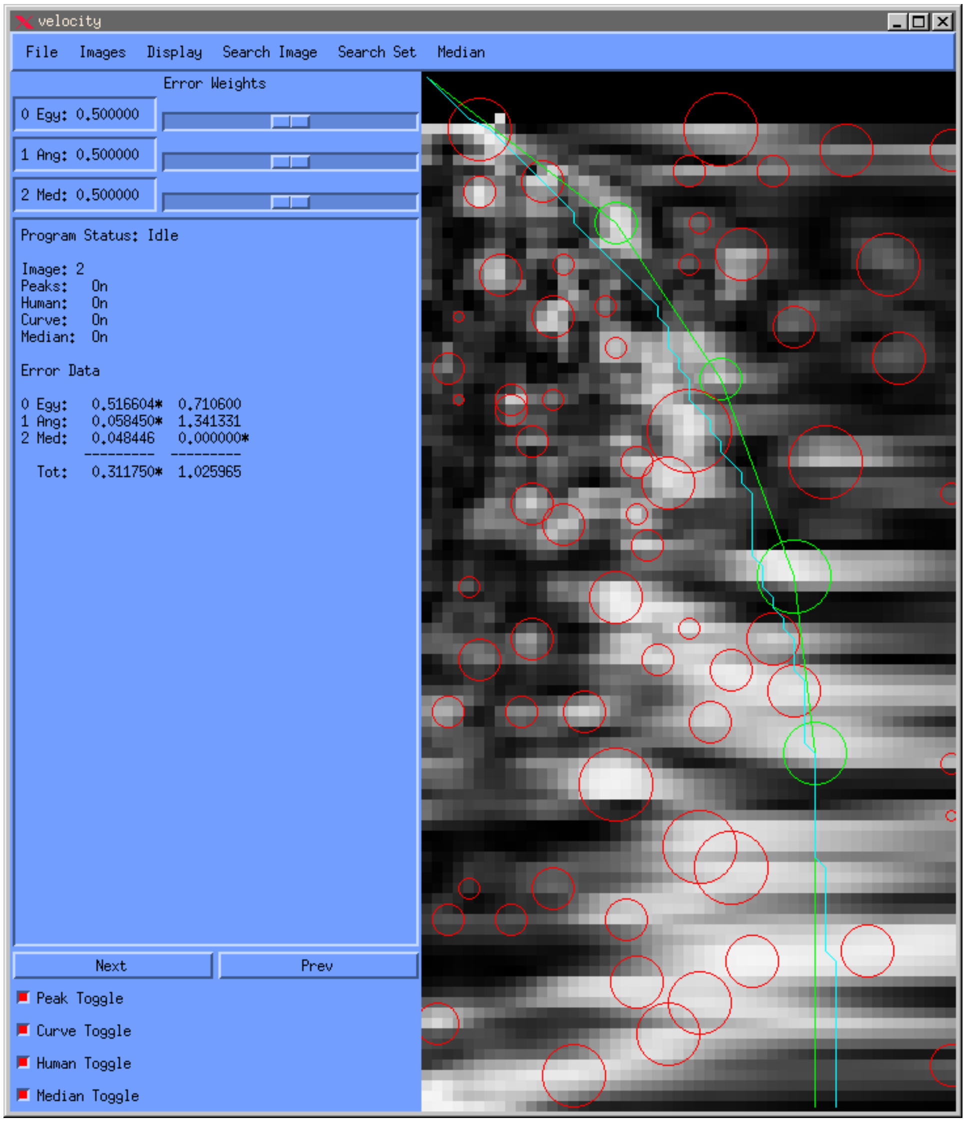


Figure 7: The GUI

being added, the system can ensure that segments with negative slope that go to the left are not allowed.

After a great deal of experimentation, we ended up using rather simple hill climbing methods for search. There are three main methods of search. Using a "steepest ascent hill climbing search," every detected peak (i.e. those not assigned a value of $ALWAYS$ or $NEVER$) is toggled either ON or OFF, one at a time, and the resulting strings are evaluated. The peak that results in the largest decrease in the error (i.e. it has the highest energy and best fits the constraints) is found and toggled. The process continues until there are no more improvements possible. The act of "flipping a bit" can sometimes have side effects. For example, when a peak is turned on, all peaks above and to the right of it must be turned off, as must all peaks below and to the left of it. This constraint keeps the string within the set of valid solutions. Since the addition of a peak can cause other peaks to be removed, the space of curves is connected in a way that allows neighboring strings to differ by more than one cell.

The first of the three available methods of search is called the "Zero Single Search." This search method currently appears to hold the most promise. To begin, any bits which are set to ON are flipped to OFF, and a single pass of hill climbing is applied. This search works in a similar way to what is knows as a "greedy tree search." The polyline can be though of as a tree, where the root of the tree is the first peak that is selected. Note that when this first peak is selected, the problem is then decomposed into two subproblemsr: first the set of peaks in the top half of the image and the set of peaks in the bottom half. The algorithm flips the bit that gives it the most improvement at each decision. This process continues until no further improvement can be made. In effect this is a single pass bit climber that starts from the string of all zeros and climbs to a local optimum.

The second search method is called the "Current Single Search." This search is identical to the Zero Single, except that the string is not first set to OFF as it is in the Zero Single Search. "Current Single Search" is used by the user to search the neighborhood surrounding the current candidate solution for improvements. The "current" solution that is used as the start state of the search can be defined by the user or randomly selected.

The third search method is the "Multiple Search." This search begins with a Current Single and a Zero Single search, followed by j trials of random restart local search. For these j trials, solutions are generated at random and then hill climbing is applied to each. Of the resulting $j+2$ (including Zero Single and Current Single) locally optimal solutions, the best solution is chosen.

The automated method we have developed uses "Zero Single Search." In practice, we have found this method generates solutions closer to those generated by humans. Since it is biased toward the all-zero string by the starting conditions, it tends to find good solutions that also uses a minimun of peaks.

3.7.1 Error Terms

A solution is evaluated using several criteria. Within the program, this translates to an evaluation function with multiple error terms. An Error Term is a measure of how good a soluton is as measured by one distinct criterion. For example, one error term rates the smoothess of the curve. Values of an error term fall in or near the range $[0, 1]$. A final combined error is then computed from the error terms using a weighted sum. The weights for the sumation (which fall in the range $[0, 1]$ for each of the error terms) are obtained from the user. By using multiple error terms and computing a weighted sum across them, we can create an evaluation function that takes into account several criteria simultaneously while applying varying significance to each. We have experimented with numerous constraints and have identified three that work best.

The first, and most obvious, is the "Energy Error Term." As we stated earlier, the total energy under the curve is one measure that has geologic significance. Since we have pre-computed the energy under any conceivable segment, we simply sum up the pre-cached Energy-Length products, and divide it by the sum of the Lengths of the segments. The resulting value represents the average intensity under the curve. To turn this value into an error, we subtract it from 1. By using Energy as the only error term, we can find the solution with the most energy, however there are many solutions that are not geologically possible.

A second error term is the "Angle Error Term." It is the average of the squares of the turning angles in the polyline. This metric keeps the polyline smooth and helps to generate more realistic solutions.

The final error term, the "Median Error Term," is much more complex and, because it requires a more detailed explanation, will be discussed in the following section.

The GUI also allows the user to try other error terms we have experimented with. It also allows the user to weight the error terms or to use a default set of weights.

3.7.2 Median

The Median method is the heart of our most successfull algorithm. It relies upon two assumptions. First, the velocity between images should not differ greatly, since all of the images in a given set are taken from approximately the same geographical site. Second, using the two earlier error terms (Energy Error and Angle Error) alone, a zero single search will result in a reasonable solution the majority of the time. When it does fail, the resulting solution is clearly an outlier.

To begin, a Zero Single search is performed on the entire set of i images. The result is i separate semblance velocity picks and the associated polyline connecting the peaks. Each of these solutions specifies a particular velocity at a particular time. Therefore, at any given time we have i velocities. For each time value, the median of these i values is computed and a new artificial median semblance velocity polyline is ploted through the data. After the median curve is computed, a third error term can be used on each of the images. Its value is proportional to the area between the polyline being evaluated and the median polyline. The use of this third error term adds continuity to the solutions in the set. By using this median polyline as an exemplar, but still allowing the program to search each image individually, a trend can be preserved while still exploiting the variations in each image. The Median method can be repeated in an iterative fashion in order to refine the median if necessary. In practice, a single refinement after the initial median calculation improves the result, but further refinements produce rapidly diminishing amounts of improvement.

4 Results and Evaluation

Example outputs of our velocity picking system are given in Figures 8 and 9. Eight semblance velocity images are presented along with the set of peaks (presented as red circles), the hand picked solutions (the white line), the automated set of picks (the green line) and the median over the set of automated picks across the seismic survey (the blue line).

One thing that is striking about the median is that it is so close to the set of hand picks. On the first pass of the local search algorithm there are some solutions that are clearly wrong–they are quite distant from the hand picked solutions. But by then computing the median and weighting the search to look for locally optimal solution that are also close to the median, all solutions are now near the hand picked solutions.

Our analysis suggests two interesting facts about the set of automated solutions. First, the area between the polylines representing the median (blue line) and the line associated with the hand pick velocities (white lines) is *less than* than the area between the individual automated solutions (green lines) and the line associated with the hand picked velocities. This could be interpreted as evidence that using the median is better than using the individual solutions since it is closer to the hand-picked solutions. But median does not maximize the energy in each individual semblance image.

In fact, the energy associated with the automated solutions is greater than the energy associated with the hand picked solutions. Thus, it is possible the set of automated solutions are better than the set of hand picked solutions. A geophysicist–perhaps even one familiar with the local geology represented in these images–would have to make that determination. But it should not automatically be assumed that being closer to the hand picked solution is necessarily better.

5 Technology Transfer

Landmark was extremely helpful in defining the problem, in generating data, and in helping us to stay informed about the state of the art. This work has also reinforced out interest in finding connections between computer vision and heuristic search.

It is unfortunate that the second year of research will not be done on this system. We feel this project had the potential to have a major impact on how velocity picking is done in the geophysics industry. We originally did not expect to get so much done in one year. (In particular, we had expected to get a system working, but did not expect to have a graphical user interface on the system.) Year two would have focused more on comparative evaluation. (Landmark had planned to implement a spline based velocity picking systems for comparative purposes.) In year two we would have also delivered this technolgy to Landmark in a form compatible with their software tools and products.

At the time we began this project we knew of no automated system to solve this problem. Landmark has since learned that some major oil companies have been using splines to solve this problem. The spline (or snake) is fit through a set of adjacent semblance velocity images–which are treated as a volume. The intersection of the spline with each image is, of course, a line (in this case

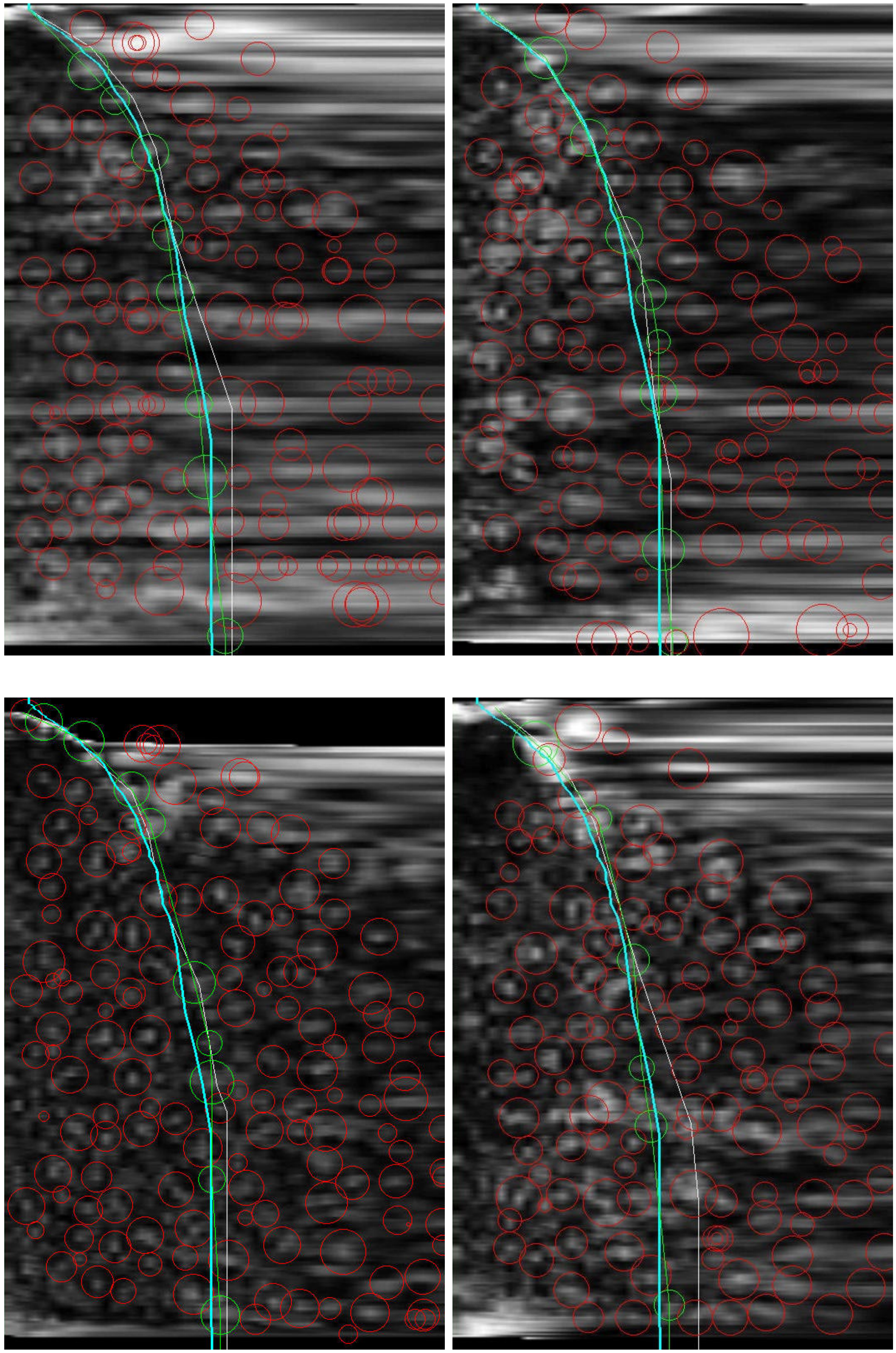


Figure 8: Sampling Solutions on a very hard dataset. The white line are hand picks. The green line the automated picking results. The blue line is the median of all picks in the seismic survey.

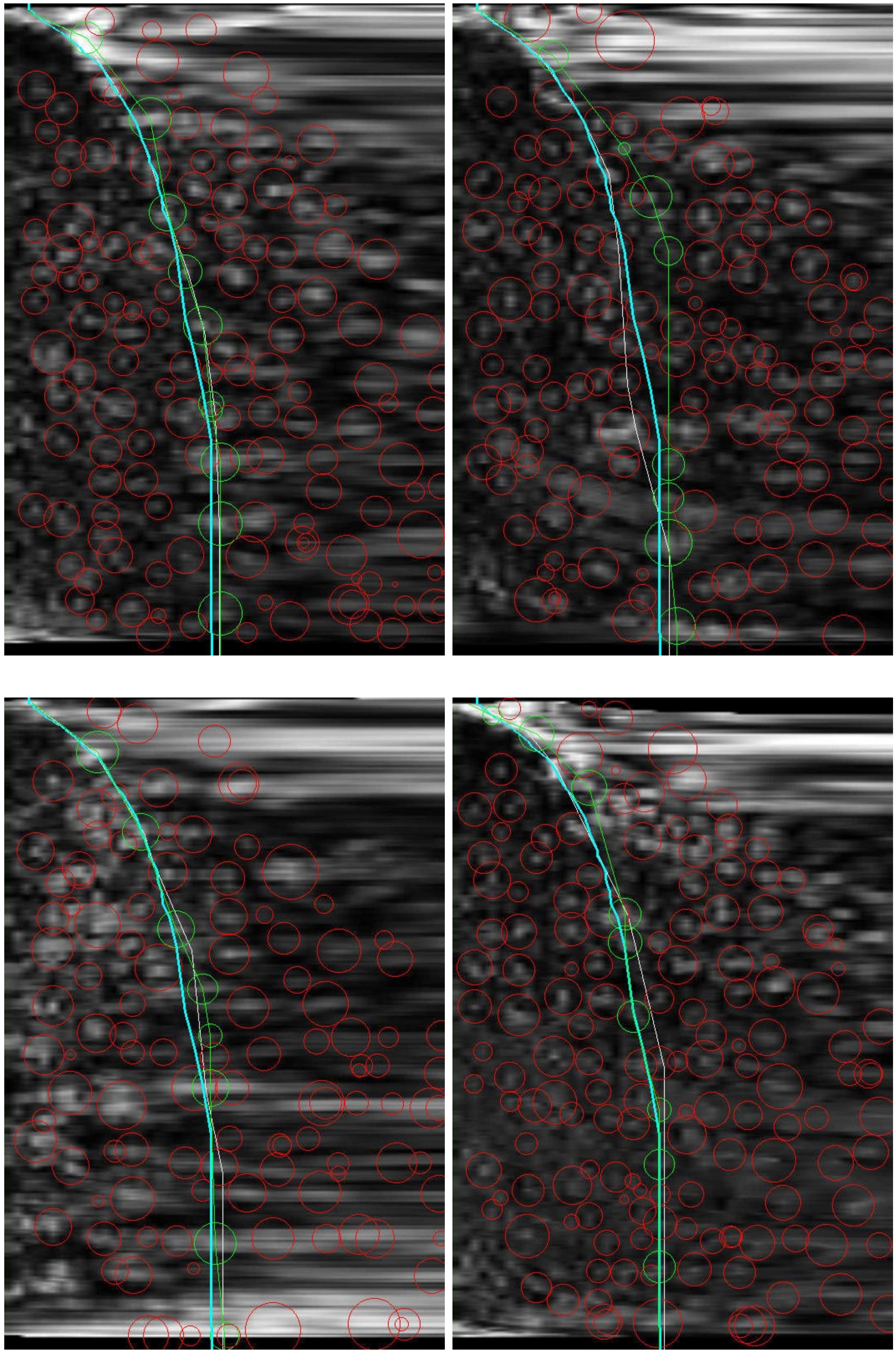


Figure 9: Sampling Solutions on a very hard dataset. The white line are hand picks. The green line the automated picking results. The blue line is the median of all picks in the seismic survey.

a curve) analogous to the polylines generated by our method. In this way, the surface generated by the spline must maximize the energy under each curve, but will obviously also be constrainted to have strong continuity from one image to the next.

We feel our method has some potential advantages compared to splines.

1. The evaluation function used by our method is much cheaper to compute and allows us to preprocess and cache a major part of the evaluation function. We utilizing only a small finite set of line segments; this allow the associated energy for each line segment to be precomputed. Since splines can move around in the semblance velocity image in a much more arbitrary fashion, it is not possible to precompute the energy component of the evaluation function.

2. Our method allows real-time interactive searches to be done due to our fast evaluation. The precomputed energy values can be generated at the same time the velocity semblance image is generated.

3. The interactive nature of our system allows a user to click on a peak in the semblance velocity image and to get a solution back consistent with that pick in real-time. Our experience suggests that picking a single peak is usually sufficient to restrict the resulting solution to the best locally optimal solution that also contains that peak. However, we have not rigorly tested this hypothesis.

5.1 Technology Transfer benefits as described by Landmark.

According to Landmark, the main advantage of this system compared to other systems is that it does allow interactive real time processing. For any automated system, users will most likely want to review the solutions to make sure they are reasonable. If a user finds a solution they do not like, they can simply pick a single peak and recompute a solution. Because of the precomputing of the evaluation associated with line segments, for the user the new solution is generated more or less immediately (e.g. within a second). This kind of interactive process is extremely valuable and is not possible with spline based methods.

6 Networking

The vision research group here as CSU is very much interested in this application. We are colloborating with that group in an effort to use the velocity picking problem as an example of how computer vision methods can be applied to real world applications. Related work was also presented at AAAI-98 in Madision, Wisconisn.

7 Publications

Our results have been recent obtained and we are only now starting to produce papers. A paper is being submitted to the *Computer Vision and Pattern Recognition Conference (CVPR)*. We also hope to publish some of this work in the geophysics community.

Bootstrap Aggregation with Medians B. Draper, K. Baek, C. Ross and J.R. Beveridge Submitted to *Computer Vision and Pattern Recognition Conference (CVPR).*

8 Funding Attributable to CASI support

Dr. Whitley has a grant submitted to NSF and Dr. Beveridge expects to submit a grant to NSF in early 1998. Dr. Whitley's proposal is currently under review.

References

[1] J. Ross Beveridge. *Local Search Algorithms for Geometric Object Recognition: Optimal Correspondence and Pose.* PhD thesis, University of Massachusetts at Amherst, May 1993.

[2] J. Ross Beveridge and Edward M. Riseman. Optimal Geometric Model Matching Under Full 3D Perspective. *Computer Vision and Image Understanding*, 61(3):351 – 364, 1995. (short version in IEEE Second CAD-Based Vision Workshop).

[3] J. Ross Beveridge, Edward M. Riseman, and Christopher Graves. Demonstrating Polynomial Run-Time Growth for Local Search Matching. In *Proceedings: International Symposium on Computer Vision*, pages 533 – 538, Coral Gables, Florida, November 1995. IEEE PAMI TC, IEEE Computer Society Press.

[4] J. Ross Beveridge, Rich Weiss, and Edward M. Riseman. Combinatorial Optimization Applied to Variable Scale 2D Model Matching. In *Proceedings of the IEEE International Conference on Pattern Recognition 1990, Atlantic City*, pages 18 – 23. IEEE, June 1990.

[5] D. M. Nguyen. An iterative Technique for Target Detection and Segmentation in IR Imaging Systems. Technical Report November, (CECOM) Center for Night Vision and Electro-Optics, 1990.

[6] B. Fish. A Neural Network Approach to Automate Velocity Picking. In *Proceedings of SEG* 64^{nd} *Annual International Meeting.* Society of Exploration Geophysicists, 1994.

[7] Pascal Fua and Yvan G. Leclerc. Model driven edge detection. In *Proceedings: Image Understanding Workshop – 1988*, pages 1016 – 1021. DARPA, Morgan Kaufmann, April 1988.

[8] J. Ross Beveridge, Edward M. Riseman and Christopher R. Graves. How Easy is Matching 2D Line Models Using Local Search? *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(6):564 – 579, June 1997.

[9] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *First International Conference on Computer Vision Proceedings*, pages 259 – 268, 1987.

[10] Mark R. Stevens and J. Ross Beveridge. Optical Linear Feature Detection Based on Model Pose. In *Proceedings: Image Understanding Workshop*, pages 695–697, Los Altos, CA, February 1996. ARPA, Morgan Kaufmann.

[11] Mark R. Stevens and J. Ross Beveridge. Precise Matching of 3-D Target Models to Multisensor Data. *IEEE Transactions on Image Processing*, 6(1):126–142, January 1997.

[12] George Reynolds and J. Ross Beveridge. Searching for Geometric Structure in Images of Natural Scenes. In *Proceedings: Image Understanding Workshop*, pages 257–270, Los Angeles, CA, February 1987. ARPA, Morgan Kaufmann.

[13] John Scales. *Theory of Seismic Imaging*. Springer-Verlag, Publishers, 1995.

[14] D. Whitley, R. Beveridge, K. Mathias, and C. Graves. Testing Driving Three 1995 Genetic Algorithms. *Journal of Heuristics*, 1:77–104, 1995.