Multiscale analysis for muon-scattering data

Guangliang Chen
Gilad Lerman
University of Minnesota
Rick Chartrand
Los Alamos National Laboratory

October 12, 2006

1 Introduction

1.1 Cosmic-ray muons

Cosmic rays from deep space are continuously striking the Earth's atmosphere. The result is a shower of particles of various sorts, travelling in various directions at nearly the speed of light. Some of these particles are so unstable that they decay before reaching the Earth's surface. Others are less energetic and are absorbed by the atmosphere. Muons, on the other hand, are not so unstable and have only a very weak hadronic interaction. The result is that at the Earth's surface, there is a shower of muons coming from all directions between the horizon and the zenith. These muons can pass through several feet of lead and many yards of rock. Unlike neutrinos, however, they do not make it through the entire planet, which is why muons only arrive from above. The flux is approximately one per square centimeter per minute.

Muons are similar to electrons, and have an electric charge (which may be positive or negative). Their one significant interaction with matter is Coulomb scattering, as they are deflected by electron clouds and nuclei. Their path through matter can be regarded as a form of random walk. The overall deflection angle as a muon of a given energy passes through a given thickness of material is roughly a Gaussian random variable. The standard deviation of the distribution is the number of *radiation lengths* present in the given thickness of material. The radiation length of a material is a decreasing function of both its density and its atomic number (Z). Since density also tends to go up with increasing Z, the amount that a muon will be scattered by a given thickness of material is strongly related to the atomic number of the material. This is the key to the usefulness of muon radiography, as the danger presented by materials in the context of nuclear smuggling also tends to go up with atomic number. Even non-radioactive materials such as lead or tungsten should be regarded as threatening, due to their ability to act as a radiation shield for the purposes of avoiding detection by conventional means. This is why our problem is framed in terms of detection of high-Z materials, without worrying much about discriminating among high-Z materials. The background muon radiation is perfect for this task.

1.2 The data

The data come from two sources, a mid-scale experimental detector arrangement, and simulations of a full-scale detector apparatus. The muon detectors consist of drift tubes joined together in planar arrays. Two of these detector planes are positioned above the test volume, two below. In the case of the simulations, the test volume is a rectangular volume containing an automobile. The larger the planes in relation to the test volume, the greater the fraction of muons passing through the test volume that will be detected. In the simulations we have data for, the detectors are 8 meters by 4.5 meters, with detectors in a pair spaced one meter apart, and the detector pairs 4.5 meters apart.

Each muon that is detected will then have four locations associated with it. Muons that do not pass through all four detectors are disregarded. The near-light speed of the muons combined with the relatively low flux rate allows muon detections at the four detectors to be associated with a single muon. See Figure 1.

A typical simulation is for 60 seconds, in which time about 95,000 muons are detected. The raw data consists of four locations for each of these 95,000 muons.

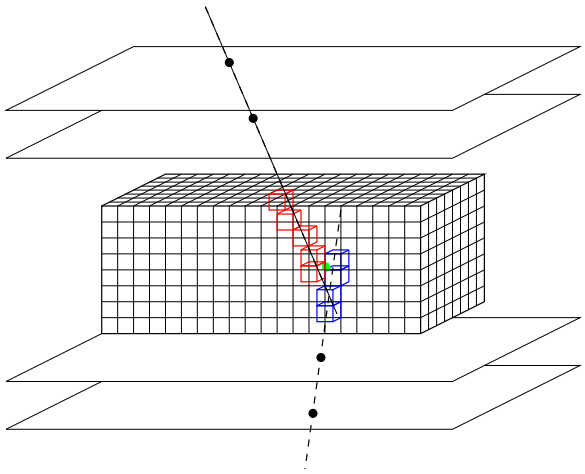


Figure 1: Entry and exit rays and the voxels they pass through are shown. The point of closest approach is green.

1.3 Preliminary processing

Each of the two pairs of locations for each muon define a line in space. We regard these as an entry ray and an exit ray. We compute the angle between these rays, and regard this as the total scattering. An experimentally challenging issue is to determine the energy of each muon. At best, this can only be estimated; our simulations assume an energy estimate with a relative error of about 29%. The scattering angle is then multiplied by the energy estimate in order to normalize for energy. This product is subsequently referred to as the scattering for the muon. This represents the best possible estimate, given the data, of the nature of the material that the muon scattered through. Of course, a single muon can have a large scattering as a result of passing through a small amount of high-Z material, a large amount of medium-Z material, or as a result of a chance fluctuation. It is only in the aggregate of many muons that meaningful results are possible.

The next step is to estimate where the scattering may have occurred. This will be at best a coarse estimate, as the data do not allow us to distinguish between single and multiple deflections. The best location estimate is the point that is closest to the entry and exit rays (see Figure 1). This point will be uniquely defined unless the rays are exactly parallel, a rare event. We call this point the point of closest approach, or POCA. This represents a single-scatter approximation of the muon's path.

The simplest reasonable reconstruction is simply a plot of POCAs, perhaps colored according to the magnitude of the scattering. For simple scenes, even this is enough to distinguish the presence of high-Z material with data collected for about 60 seconds. However, it is important to be able to identify the presence of high-Z material when significant amounts of medium-Z material are present in any arrangement, and to do so with very high reliability and with as little data as possible. All methods improve with collection time, as the muon flux is the limiting factor. However, it will be crucial to have a method that can operate at a border crossing without causing delays.

For computational reasons, locations are generally defined discretely, in terms of voxels. We seek to identify not the location of high-Z material, but the voxel in which it is contained. As before, the best estimate of the voxel in which a muon has scattered is the one that contains the POCA. One can then regard each voxel as a histogram bin, and tally up scatterings of muons having their POCAs in a given voxel. The result is a three-dimensional histogram of scattering. We also compute a histogram of muon count and of squared scattering. With these, any of several statistics can be computed. The one that seems to work best is mean squared-scattering.

1.4 Maximum likelihood reconstruction

A method for estimating the scattering propensity in each voxel that is very different from the POCA reconstruction method is to compute a maximum likelihood estimate. The description of this method is beyond the scope of this report; see [6]. The ML reconstructions used in the work described here were computed by Larry Schultz.

1.5 Voxellation and neighborhoods

An unsettled issue is one of voxel size. The smaller the voxel size, the greater the resolution, but the fewer the number of muons that will pass through each voxel. Also, even when one is seeking to identify roughly voxel-sized objects, the signal may be spread over several voxels. Currently, we use 10-centimeter voxels, though we often work not with individual voxels but with 27-voxel neighborhoods. How this differs from using 30-centimeter voxels has not been explored.

The result of this processing is that each voxel has a point in 27-dimensional space associated with it, with the components being the statistics for the voxels in the neighborhood. Much work has been done using this 27-dimensional data to train a machine-learning classifier to tell whether high-Z material is present in the central voxel. After early success, this method has had mixed results. More importantly, it must be coupled with some other method to tell which locations should be tested with the classifier.

1.6 Data clouds

The idea explored in this report is to consider the cloud in 27-dimensional space consisting of all the 27-voxel neighborhoods in the test volume. Preliminary studies suggested that these clouds will differ in geometric structure depending on whether high-Z material was present in the volume. See Figures 2, 3, and 4. However, the differences can be subtle and difficult to characterize. Nevertheless, the value of even moderate success in distinguishing high-Z material clouds would be substantial.

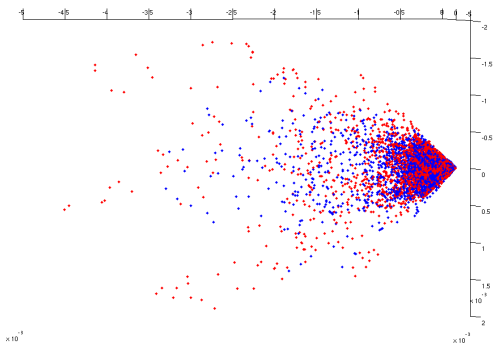


Figure 2: Data clouds for each of two automobile simulations. The red cloud comes from a simulation with 20-kg uranium spheres present, the blue comes from one with no high-Z material. Shown is one view of the projections of the clouds onto the first three principal components. Differences should be greater in the native 27-dimensional space. The red cloud is slightly longer in this view, but otherwise little difference is apparent.

2 Test scenarios

The multiscale analysis was carried out on three sets of data, which we describe in this section.

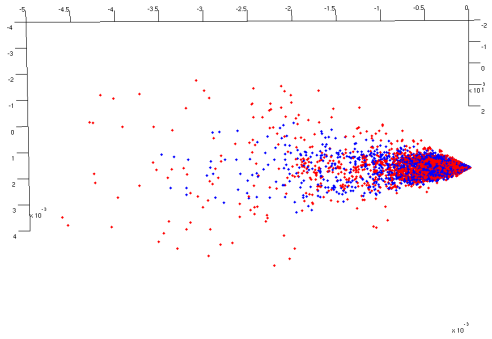


Figure 3: The red cloud is both longer and broader, with some filament structure present in the outliers.

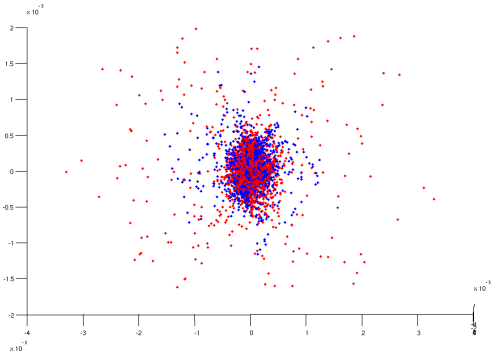


Figure 4: View from the apex of the clouds.

2.1 Simulated car scenes

In this data set there are 10 different simulated car scenes. The simulations were carried out by Alexei Klimenko, using GEANT. The scenes, labeled car 6 through car 15 for project historical reasons, are as follows:

Car 6: U cylinder in steering column, 5 cm radius, 67 cm long.

Car 7: Car only, no high-Z material.

Car 8: Cylinder oriented as in Car 6, but only 7 cm long, and has 1 cm radius U wrapped in 2.5 cm radius of Pb shielding.

Car 9: Cylinders of same radius as Car 8, but 15 cm long, and located in both bumpers of the car.

Car 10: Two spheres of U, radius 8.33 cm, one centered in the car, the other in the battery.

Car 11: Two shielded cylinders, U radius 2 cm and length 4 cm, shielded by Pb, radius 5 cm, length 10 cm, located in the axles.

Car 12: As in Car 11, but only the high-Z objects, no car.

Car 13: As in Car 10, but radius now 6.33 cm.

Car 14: As in Car 13, but with lighter wheels.

Car 15: As in Car 7 (no high-Z), but with lighter wheels.

Ten simulations representing 60 seconds of muon exposure were produced for each scene. The data were processed to produce a POCA reconstruction of mean squared scattering (see Section 1.3) and a maximum likelihood reconstruction (see Section 1.4).

2.2 More simulated cars

This data set consisted of slight variations of the last two scenes described above, cars 14 and 15, with the former having high-Z material and the latter not. Slightly over 1000 simulations of 60-seconds each were run by Matthew Sottile. In this case, only POCA reconstructions of mean squared scattering were computed.

2.3 Experimental data

The experimental data come from four event files from the current mid-scale apparatus at LANL. Each consisted of about $170,000$ muon tracks. For two of the samples, the test volume contained a tungsten cylinder, and the word "LANL" spelled out in 1-inch lead stock. For the other two samples, the test volume was empty. POCA reconstructions of mean squared scattering were computed.

3 Our model for the data

We assume that the values of the muon-scattering intensities in voxel neighborhoods have been generated as a mixture of two distributions. The first one (the "stable" part) represents the background muon scattering, whereas the second one (the "outlying" part) represents the muon scattering due to high-Z material. We suppose that the vectors of voxel neighborhoods arising due to the stable distribution are concentrated around a Lipschitz graph (see Subsection 4.1). More precisely, we assume that the conditional expectation (according to the stable distribution) of $\mathbf{y}$ given x is a Lipschitz function which we denote by $\mathbf{C}(x) = (C_2(x), \ldots, C_D(x))$ for the given data (in $\mathbb{R}^D$). For example, the very special case when the curve is the line spanned by the uniform vector $(1, \ldots, 1)'$ represents uniform values among voxels of the stable distribution (background scattering). We further assume that the values of voxel neighborhood sampled according to the stable distribution are distributed normally around that curve. That is, $\mathbf{y}|x$ (sampled according to the stable distribution) is normal with mean $\mathbf{C}(x)$ and with radial symmetry (the distribution is determined by $\|\mathbf{y} - \mathbf{C}(x)\|$). We denote by $S(x)$ the corresponding conditional standard deviation and assume that S is finite everywhere.

4 The MSC algorithm

The MSC (short for Multiscale Strip and Curve construction) algorithm was first developed in [5, 4]. It assumes that a given dataset satisfies the model suggested above, and it constructs a normalizing curve (the estimated conditional mean) with an enveloping strip (the estimated conditional variance) around it in a multiscale fashion. The conditional variance estimate is used to identify outliers.

The algorithm builds on the following intuition: it is possible to approximate the data set by lines with cylindrical regions around them at various locations and scales. Those lines are combined together in order to estimate the underlying Lipschitz graph (of the function $\mathbf{C}$). The cylindrical regions exclude "initial outliers." Inside them the algorithm estimates the local standard deviation S and then uses this estimate together with the one for $\mathbf{C}$ in order to reassess "outliers" of the whole data set.

We sketch our algorithm below (Algorithm 1) for the case $n_{sh} = 1$. Later subsections contain the details of the different steps of this scheme and the case where $n_{sh} > 1$. In this pseudocode and throughout this section we denote our underlying data set by E, N the number of points in E, and D the ambient dimension. We fix a line L that approximates E globally. We also assume that the data has been transformed orthogonally so that L coincides with the first coordinate axis.

The algorithm depends on the following parameters: l_0, c_0, n_0, n_{sh}, and α_0. However, the choices for their values are not arbitrary; the optimal values are selected in a manner to be briefly discussed later, and elaborated further elsewhere [5].

Algorithm 1 MSC algorithm

- Approximate E by a line L, and apply a rigid transformation to E, so that $L = x$-axis
- Set $Q_0 :=$ interval containing the projection of E onto L
- Set $\ell = 0$, $Stop\_Int = \emptyset$, $Good\_Int = \{Q_0\}$, $N_{\text{stop}} = 0$

while $l \leq l_0$ and $N_{\text{stop}} < N$ **do**

- For each interval $Q \in Good\_Int$ compute line L_Q and form a cylindrical region $Cyl(Q)$ around it
- Compute f_Q which describes a local fraction of putative "outliers" inside $Cyl(Q)$
- Compute F_Q which combines f_P's for intervals P's from current and previous levels containing Q
- Compute σ_Q: standard deviation in $Cyl(Q)$
- $New\_Stop :=$ all intervals in $Good\_Int$ satisfying: $F_Q > \alpha_0$ or $|Cyl(Q)| < n_0$.
- $Stop\_Int := Stop\_Int \cup New\_Stop$
- $Good\_int :=$ dyadic children of intervals in $Good\_int$
- $Good\_int := Good\_int \setminus Stop\_int$
- $N_{\text{stop}} :=$ number of points in stopping intervals
- $l := l + 1$

end while

- Record local standard deviations for stopping time cylindrical regions to get $\tilde{S}$
- Record local line approximations for stopping time cylindrical regions to get $\tilde{\mathbf{C}}$
- Obtain ranking $\tilde{R}$ by using $\tilde{S}$ and $\tilde{\mathbf{C}}$
- Identify outliers according to $\tilde{R}$

4.1 Some notation and definitions

If K is a subset of $\mathbb{R}^D$, we denote by $|K| \equiv |K \cap E|$ the number of points of E in K. We use the notation $\overrightarrow{\mathbf{u}} = (x, \mathbf{y})$ for a point in $\mathbb{R}^D$; that is $\mathbf{y} = (x_2, \ldots, x_D) \in \mathbb{R}^{D-1}$. We denote by $\|\overrightarrow{\mathbf{u}}\|_2$ the Euclidean norm of $\overrightarrow{\mathbf{u}}$. The Lipschitz norm of a function $\mathbf{f} : \mathbb{R} \longmapsto \mathbb{R}^{D-1}$ is a global bound on the "approximate derivative" of $\mathbf{f}$ and is defined as follows:

$$\|\mathbf{f}\|_{\text{Lip}} \equiv \max_{x_1 \neq x_2 \in \mathbb{R}} \frac{\|\mathbf{f}(x_2) - \mathbf{f}(x_1)\|_2}{|x_2 - x_1|}.$$

A function $\mathbf{f} : \mathbb{R} \longmapsto \mathbb{R}^{D-1}$ is Lipschitz if and only if $\|\mathbf{f}\|_{\text{Lip}} < \infty$. The Lipschitz graph associated with $\mathbf{f}$ has the form: $\Gamma = \{(x, \mathbf{y}) | x \in [a, b] \text{ and } \mathbf{y} = \mathbf{f}(x)\}$.

4.2 Dyadic intervals and rectangles

We fix an interval Q_0 of nearly minimal length containing the projection of the data set onto the x-axis. A dyadic interval with respect to Q_0 is an interval that occurs when dividing Q_0 recursively in halves. It is of level l, if it has been obtained by l consecutive partitions. We denote all dyadic intervals with respect to Q_0 by $\mathcal{D}(Q_0)$. If Q is a dyadic interval, we denote its length by $\ell(Q)$. If $Q \in \mathcal{D}(Q_0) \setminus \{Q_0\}$, then denote by P_Q the dyadic parent of Q according to the grid $\mathcal{D}(Q_0)$ and also define $P_{Q_0} := Q_0$.

In order to describe the stopping constructions more formally, we will need to define properties of several regions, which extend dyadic intervals to the space $\mathbf{R}^D$. More details of these constructions are in [5].

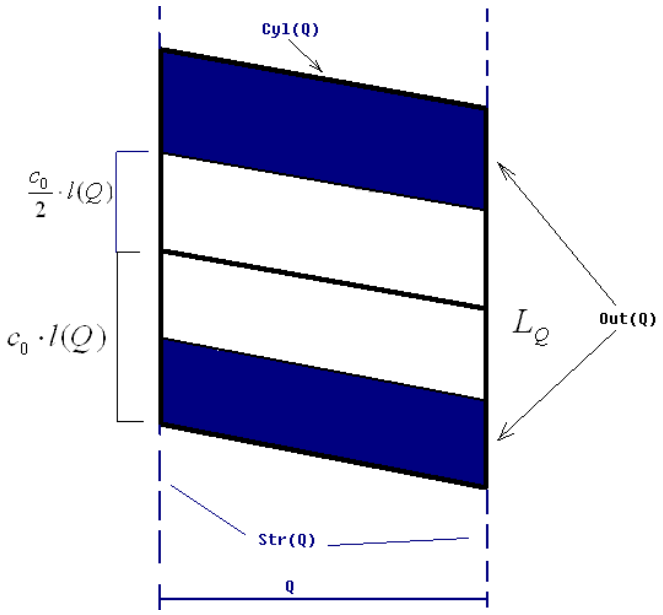


Figure 5: Demonstration of the different regions associated with the interval Q when $D = 2$.

For an interval $Q \in \mathcal{D}(Q_0)$, its extension $Str(Q)$ to an infinite strip is

$$Str(Q) = Q \times \mathbb{R}^{D-1}.$$

Its extension $Cyl(Q)$ to a cylindrical region (centered around L_Q) is

$$Cyl(Q) = Q \times \{\mathbf{y} \in \mathbb{R}^{D-1} \mid \|\mathbf{y} - L_Q(x)\|_2 \leq c_0 \cdot \ell(Q)\}.$$

The "outer" part of $Cyl(Q)$ is defined, by removing a sub-cylinder of appropriate radius, as follows:

$$Out(Q) = Cyl(Q) \setminus (Q \times \{\mathbf{y} \in \mathbb{R}^{D-1} \mid \|\mathbf{y} - L_Q(x)\|_2 \leq \frac{c_0}{2} \cdot \ell(Q)\}).$$

4.3 The stopping criterion

We briefly describe the formal steps of the stopping construction and then explain their motivation. More details of the stopping construction and the implied properties are formulated and proved in [5].

The algorithm proceeds in a top-down procedure and computes f_Q and F_Q at any dyadic interval Q it visits. The fraction f_Q has the form

$$f_Q = \frac{|Out(Q)|}{|Str(Q)|}.$$

The cumulative sum of fractions, F_Q, is computed as follows: First, the algorithm initializes $F_{Q_0} = 0$, then it applies the reduction formula (from coarse levels to fine levels):

$$F_Q = F_{P_Q} + f_Q.$$

While proceeding from top to bottom levels, the algorithm stops at an interval $Q \in \mathcal{D}(Q_0)$ (together with all of its descendants in $\mathcal{D}(Q_0)$) if any one of the following two conditions is satisfied:

$$1. \quad F_Q > \alpha_0. \tag{1}$$
$$2. \quad l > l_0. \tag{2}$$
$$3. \quad |Cyl(Q)| < n_0. \tag{3}$$

The main stopping criterion (equation (1)) implies a global estimate on the percentages of initially detected outliers (points outside the rectangular regions) as a function of the parameter α_0 [5, Proposition 5.1]. The second stopping criterion (equation (2)) controls the maximum level that the program can reach. The last one (equation (3)) is necessary for having valid local estimates in each interval.

The stopping construction results in local rectangles which aim to cover most of the "stable" set and to separate away "significant" outliers. The heuristic justification for the success of this separation can be given as follows. The local quantity f_Q measures the local fraction of "putative outliers" in $Cyl(Q)$. High values of f_Q, occurring in combination with sufficiently farther local distance from the core of the "stable" distribution, imply presence of locally significant outliers. In order to identify outliers that are also globally significant, we follow several strategies commonly used in harmonic analysis, which combine local quantities at different scales to identify global structure. We use an additive function F_Q, whose analogs have appeared in similar formulations, as for example in [2] and [3].

4.4 The output functions

The main output functions $\tilde{\mathbf{C}}$ and $\tilde{S}$, estimate the local "means" and "standard deviations" of the stable distribution. That is, in each stopping interval Q,

$$\tilde{\mathbf{C}} = L_Q, \text{ and } \tilde{S} = \sigma_Q,$$

where

$$\sigma_Q \stackrel{\text{def}}{=} \left(\frac{1}{|Cyl(Q)|} \sum_{(x,\mathbf{y}) \in Cyl(Q) \cap E} |y - L_Q(x)|^2 \right)^{\frac{1}{2}}.$$

If the number of points in Q is less than n_0, then it assigns L_{P_Q} and σ_{P_Q}. Equivalently, if $Q(x)$ is the unique stopping interval containing x, then the algorithm forms the following piecewise linear function $\tilde{\mathbf{C}}$:

$$\tilde{\mathbf{C}}(x) = \begin{cases} L_{Q(x)}, & \text{if } |Q| \geq n_0; \\ L_{P_Q(x)}, & \text{otherwise.} \end{cases}$$

and piecewise constant function $\tilde{S}$:

$$\tilde{S}(x) = \begin{cases} \sigma_{Q(x)}, & \text{if } |Q| \geq n_0; \\ \sigma_{P_Q(x)}, & \text{otherwise.} \end{cases}$$

We create a smoother version of the above functions by generating n_{sh} instances of the corresponding piecewise linear/constant functions according to different grids and averaging those piecewise linear/constant functions (see [5, Section 4.6]).

The function $\tilde{S}$ estimates "standard deviations" in the rectangles associated with stopping intervals, and requires a small correction to extend it to the region outside those rectangles. We thus alter it by assuming that for each stopping interval Q, the points in $Cyl(Q)$ were sampled from the restriction of a Gaussian random variable to that region. The function $\hat{S}$ estimates the standard deviations of the underlying local Gaussian distributions (see [5, Section 4.5]). Note that except in this last stage, the algorithm need not make any assumptions about the exact nature of the statistical distributions of data.

4.5 Ranking and identification of outliers

In the symmetric case, we can assign rankings $\tilde{R}$ and $\hat{R}$ to any point $(x, \mathbf{y}) \in E$ by

$$\tilde{R}(x, \mathbf{y}) = \frac{\|\mathbf{y} - \tilde{\mathbf{C}}(x)\|_2}{\tilde{S}(x)} \text{ and } \hat{R}(x, \mathbf{y}) = \frac{\|\mathbf{y} - \tilde{\mathbf{C}}(x)\|_2}{\hat{S}(x)}.$$

We mainly use the ranking $\tilde{R}$ due to convenience and the fact that $\tilde{R} \cong \hat{R}$ in practice.

We may fix a threshold level λ and identify a corresponding set of outliers containing all points with $\tilde{R} > \lambda$. We may also assign a p-value to any point $(x, \mathbf{y}) \in E$. That is,

$$p\text{-val}(x, \mathbf{y}) = \text{erfc}\left(\frac{\tilde{R}(x, \mathbf{y})}{\sqrt{2}}\right) = 1 - \text{erf}\left(\frac{\tilde{R}(x, \mathbf{y})}{\sqrt{2}}\right),$$

where

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \text{ for any } z \in \mathbb{R}.$$

Following Reiner et al [1], we can have adjusted p-values to control the false discovery rate of the multiple testing procedure. That is, given a false discovery rate level q, we order the computed p-values: $p_{(1)} \leq \ldots \leq p_{(N)}$ and set

$$p^* = p\text{-value}(\max\{i : \ p_{(i)} \leq q \cdot \frac{i}{N}\}). \tag{4}$$

We identify the points with p-values less than or equal to p^* as being outliers.

4.6 Choice of parameters

We fix the values of the following parameters, except α_0, as follows: $l_0 := 12$, $n_0 := 10$, $n_{sh} := 10$, and c_0 is (a constant multiple of) the ratio of the maximum L_2 distance to the length of the projection onto E.

The parameter α_0 is crucial for good performance of the algorithm. It describes the global expected percentage of outliers. We have developed an algorithm for estimating this parameter [5, Section 4.8]. The main idea is to apply the MSC algorithm with different values of α_0 and identify outliers at different thresholds on p-values. For each value of α_0, we draw the curve of the number of outliers detected by the algorithm as a function of the threshold on p-values. The "correct" value of α_0 (representing global percentage of outliers) is detected by the largest jump between the curves. However, we have skipped this step when processing the data in order to save time and have determined that $\alpha_0 = 0.1$ after trying out a few values of α_0 (e.g., 0.05, 0.10, 0.15, etc.).

4.7 Complexity

The speed of the algorithm for a data set of N points, when using ℓ_0 levels and n_{sh} shifts, is of order $O(N \cdot D \cdot \ell_0 \cdot n_{sh})$ and the required storage is $O(N \cdot D)$ (see [5, Section 5.3]). In practice, the CPU time of our algorithm was about 0.51 seconds using data with $N = 4950$ points in $\mathbb{R}^{27}$ and $n_{sh} = 1$, and an IBM T43 laptop with Intel Pentium M processor 2.13 GHz and 1 GB of RAM.

5 Numerical results and discussions

We applied the MSC algorithm to the three data sets described in Section 2 and report our results in this section.

5.1 Simulated car scenes

5.1.1 POCA data

Recall that for each car scene, the data consist of 10 samples of 4950 27-voxel neighborhoods. We applied the MSC algorithm to the matrices of 4950 points in $\mathbb{R}^{27}$. We have combined information from the ten different replicates (samples) in two different ways.

First, in order to compensate for random effects, we have computed the median ranks $\tilde{R}$ along the 10 samples. The larger the rank of a point, the greater the possibility that it is an outlier. Figure 6 shows sorted values of the ranks $\tilde{R}$ for all 10 cars.

We can see that the cars 6, 10, 12, 13, and 14 all have a significant amount of sample points that have ranks greater than 2. Cars 7, 8, and 9 have maximum ranks less than 2. Cars 11 and 15 have just a very small number of points (about 5) that have ranks slightly greater than 2. Note that $\tilde{R} = 2$ corresponds to a p-value $erfc(\tilde{R}/\sqrt{2}) = 0.0455$. Thus an initial conclusion, which may be statistically imprecise, is that at the 0.05 confidence level, we can ascertain that cars 6, 10, 12, 13, and 14 have high-Z material inside and cars 7, 8, and 9 do not. Cars 11 and 15 are a bit on the edge, thus we cannot make a decision about them.

Second, in order to make our analysis more precise in this situation of multiple testing (random values at different voxels), we have maintained a false discovery rate of 0.2. We have computed for each replicate a p^*, the threshold p-value (see equation (4)), and have identified high-Z voxels as those that are detected as outliers in at least 2 (out of the 10) replicates. Cars 6, 10, 12, 13, and 14 were detected in this way as having high-Z material. Following the ground truth information on the cars, we find that we have failed to identify cars 8, 9, and 11 as having high-Z material.

In order to obtain some insight into the misclassification of those cars, we decide to observe the data structures of the 10 cars in some way.

Recall that there are 10 cars, each consisting of 10 samples. These 100 samples are all matrices of size 4950×27. We first shift these samples by their centers of mass then rotate them according to the right singular matrices so that the main axes are the x_1-axes. We then plot the second principal axis versus the first (i.e., x_1-axes) in the xy-plane. To save space and for simplicity, Figure 7 describes the 2-dimensional representations of only 10 samples, each of which is from a separate car. Note that the plots for cars 7, 9, and 11 were intentionally made to be on the same scale for comparison.

Cars 6, 10, 12, 13, and 14 generally have much more spread-out geometrical structures than the rest, making it plausible for them to be correctly detected as having high-Z material. However, there is no general pattern for the other cars, namely 7, 8, 9, 11, and 15. Look at the attached plots that were obtained by using the first samples and notice the following: the plots corresponding to cars 7 and 11 have very similar geometrical structures, which makes it very hard to distinguish between them; the plot of car 9 (which has high-Z) has the most concentrated structure of all; and the plot of car 15 (without high-Z) even looks like having a few outliers. These abnormal phenomena suggest that the exposure time may not be long enough so that the random effects may have been dominant in the muon-scattering process.

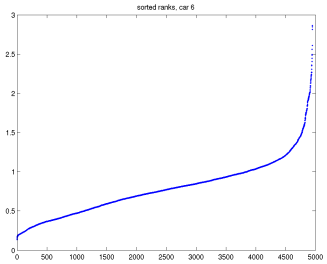
5.1.2 ML data

For this type of data, we could not obtain as good discrimination as with the POCA data. A similar two-dimensional inspection of this data is in Figure 8.

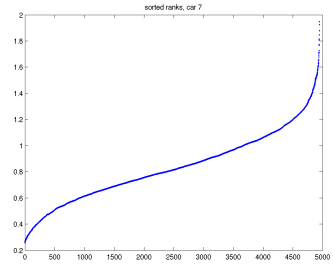
5.2 More simulated cars

We have about 1000 samples of 60-second simulations of two cars, one with high-Z material and one without. Noticing that the two settings are identical except for high-Z material being present or not, we assume the variances of the stable concentrations in both settings to be the same. The data was then processed in two different ways. One way was to take medians of ranks over several samples. The other was to first combine the data to produce histograms for longer exposure times, and then compute the ranks. The former method generally produced better results.

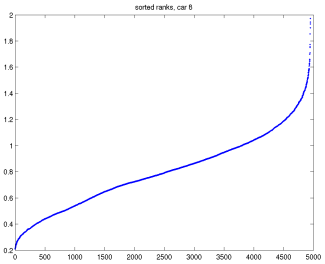
Several results are plotted in Figure 9. Both processing methods fail to distinguish well between the two cars with an effective exposure time of three minutes. With four minutes, only a few negative samples have a rank greater than the lowest 30% of the ranks of positive samples. With five minutes, this decreases to



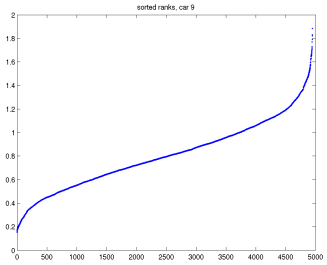
(a) Car 6



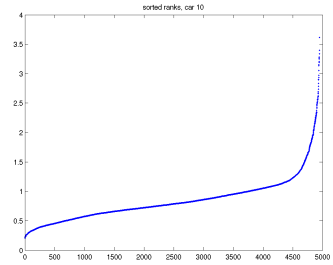
(b) Car 7



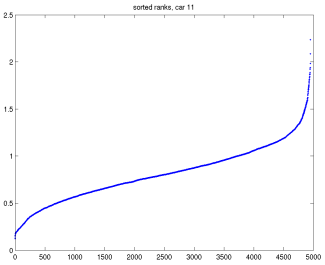
(c) Car 8



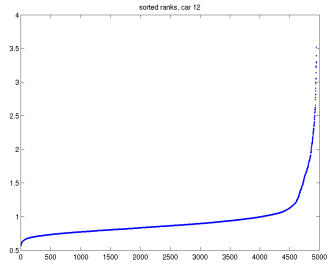
(d) Car 9



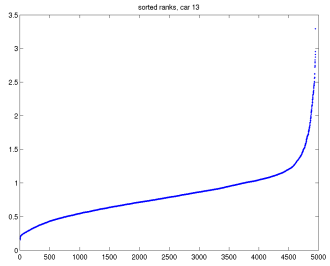
(e) Car 10



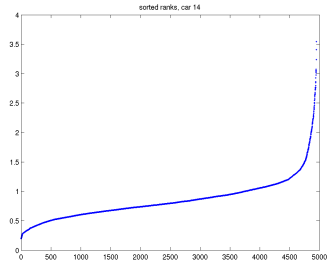
(f) Car 11



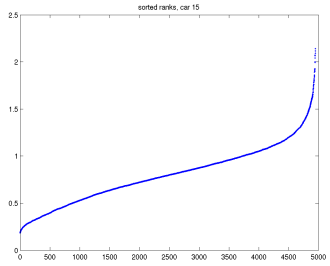
(g) Car 12



(h) Car 13

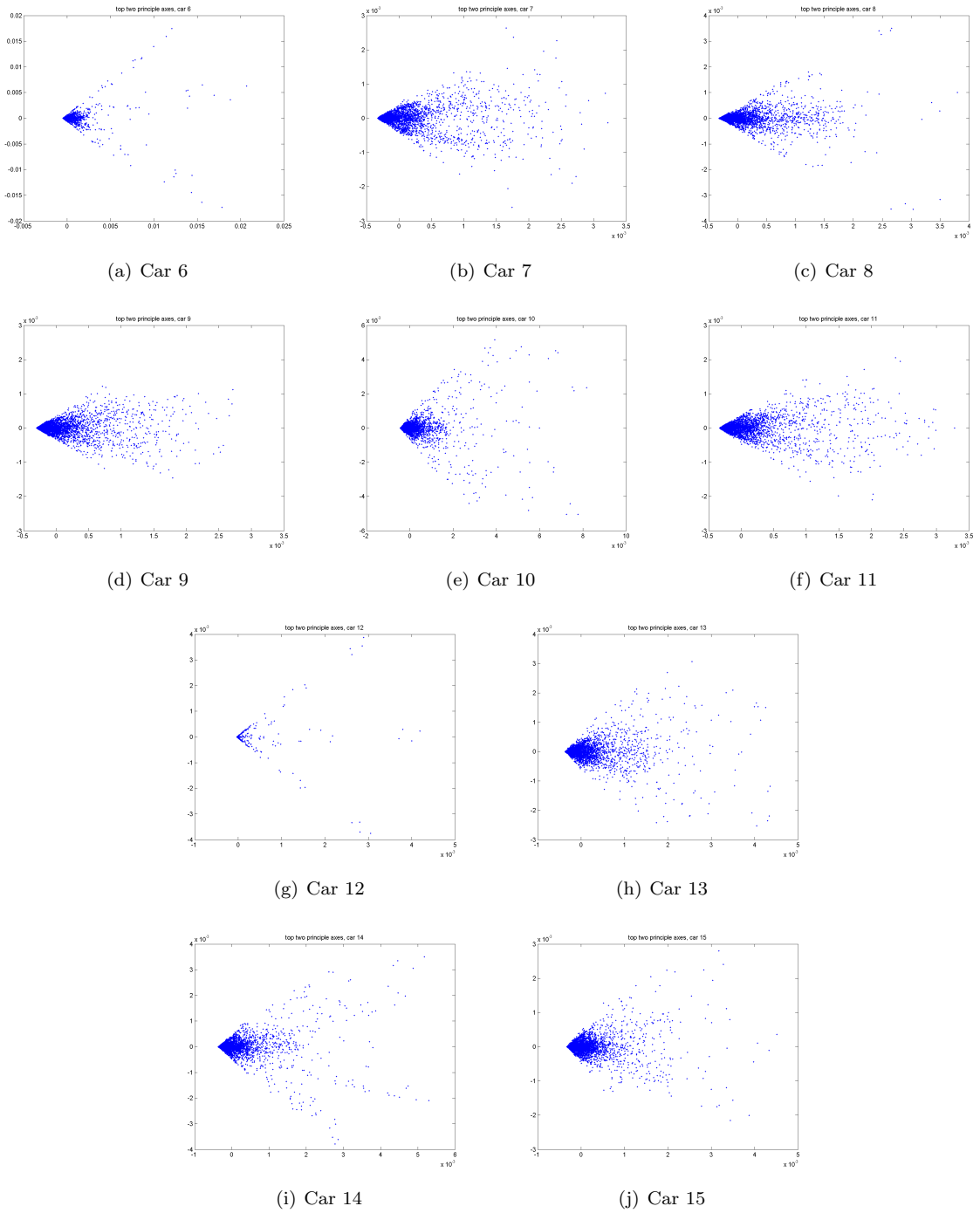


(i) Car 14



(j) Car 15

Figure 6: Sorted median rank for 10 samples for each car.



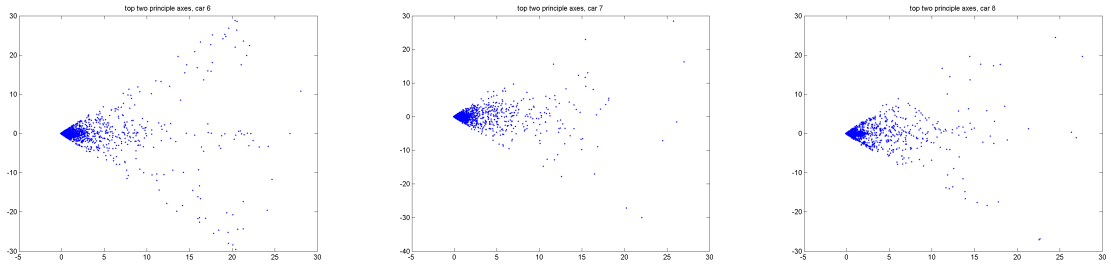
(a) Car 6 (b) Car 7 (c) Car 8

(d) Car 9 (e) Car 10 (f) Car 11

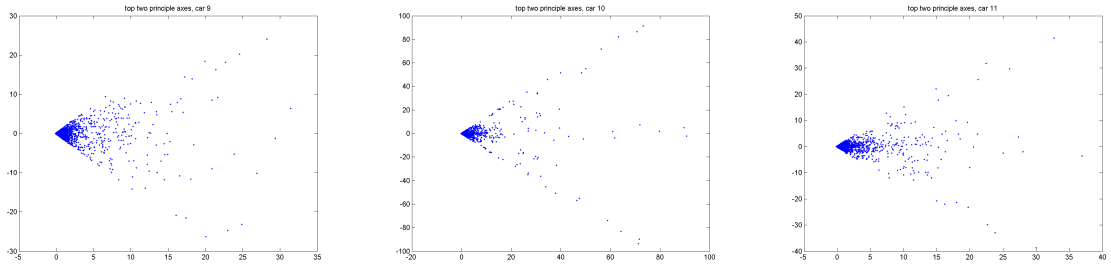
(g) Car 12 (h) Car 13

(i) Car 14 (j) Car 15

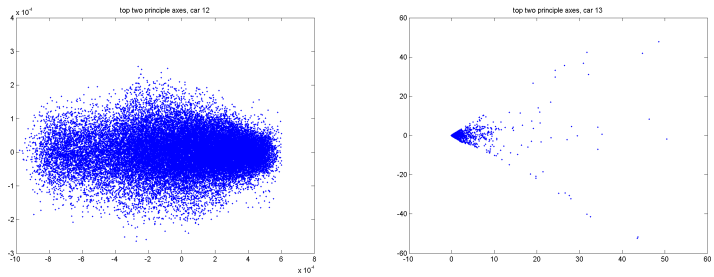
Figure 7: Plots of two-dimensional projections of the 27-dimensional voxel neighborhoods from POCA reconstructions.



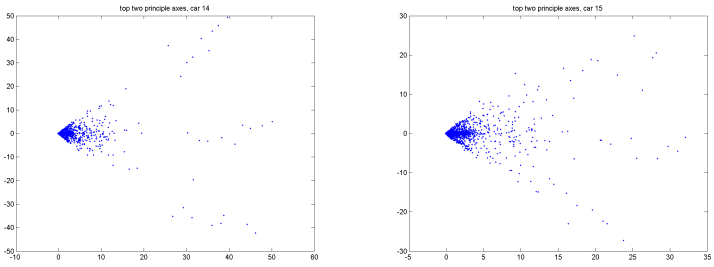
(a) Car 6 (b) Car 7 (c) Car 8



(d) Car 9 (e) Car 10 (f) Car 11



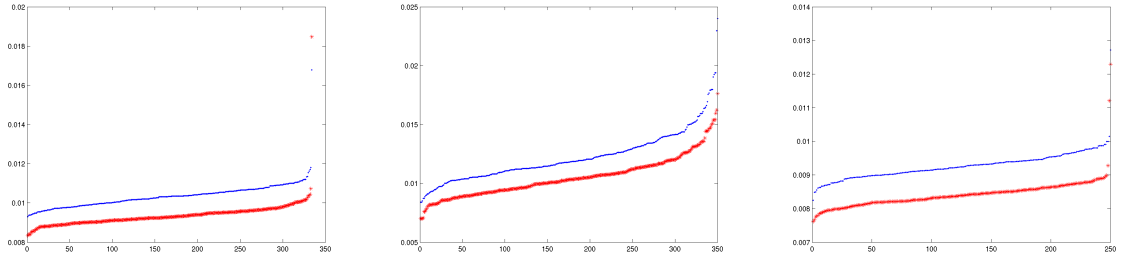
(g) Car 12 (h) Car 13



(i) Car 14 (j) Car 15

Figure 8: Plots of two-dimensional projections of the 27-dimensional voxel neighborhoods from maximum-likelihood reconstructions.

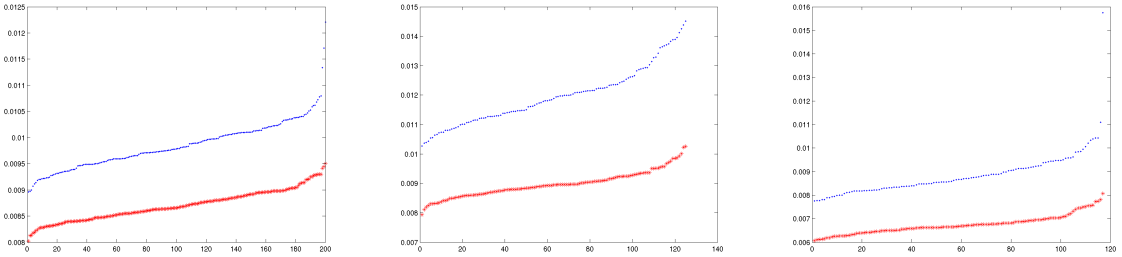
about 15%. After eight minutes, the overlap is only a few samples. Using nine minutes processed from the median of three, three-minute ranks does slightly less well.



(a) 100th-largest median rank of 3 60-second samples

(b) 50th-largest rank of 180-second samples

(c) 100th-largest median rank of 4 60-second samples



(d) 50th-largest median rank of 5 60-second samples

(e) 10th-largest median rank of 8 60-second samples

(f) 25th-largest median rank of 3 180-second samples

Figure 9: Plots of the nth largest median rank, as indicated. Blue is car 14 (high-Z), red is car 15 (no high-Z).

5.3 Experimental data

We have also applied the MSC algorithm to compute the ranks for all four samples. The plots of the sorted ranks are in Figure 10.

We see that two plots (plots 1 and 3) have highest values around 2 while the other two (plots 2 and 4) have highest values around 6. Moreover, for the latter, there are large jumps on the right ends of the plots. These are all strong evidence that experiments 2 and 4 have high-Z material present and 1 and 3 do not. This matches the true nature of the four experiments.

References

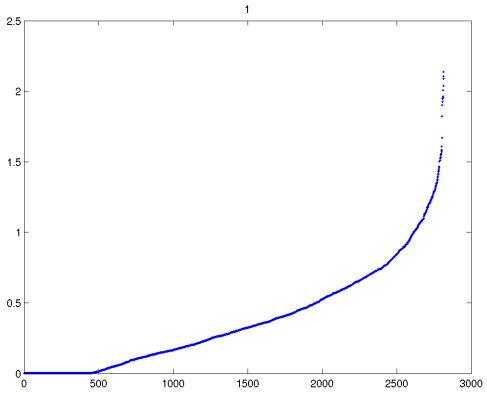
[1] Y. Benjamini and D. Yekutieli. The control of the false discovery rate under dependency. *Ann. Statist.*, 29:1165–1188, 2001.

[2] P. W. Jones. Rectifiable sets and the traveling salesman problem. *Invent. Math.*, 102:1–15, 1990.

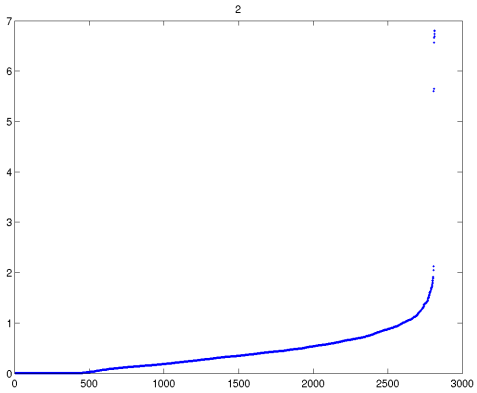
[3] G. Lerman. Quantifying curvelike structures of measures by using l_2 Jones quantities. *Comm. Pure App. Math.*, 56:1294–1365, 2003.

[4] G. Lerman, J. McQuown, A. Blais, B. D. Dynlacht, G. Chen, and B. Mishra. Functional genomics via multiscale analysis: Application to gene expression and chip-on-chip data. Submitted, 2006.

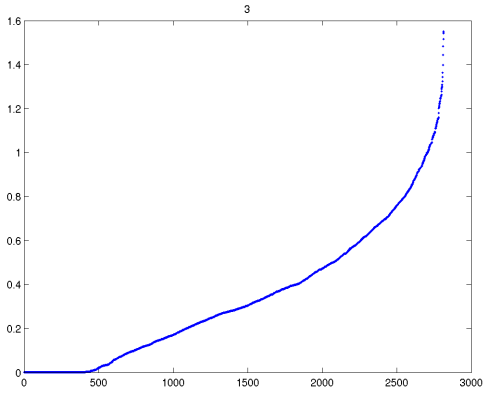
[5] G. Lerman, J. McQuown, and B. Mishra. Multiscale curve and strip constructions. In preparation, 2006.



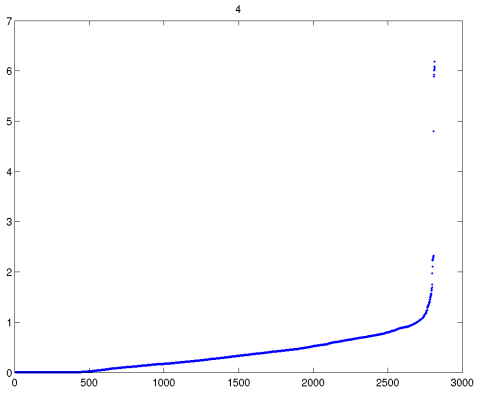
(a) Sample 1



(b) Sample 2



(c) Sample 3



(d) Sample 4

Figure 10: Sorted rank for each experimental sample.

[6] Larry Schultz, Konstantin Borozdin, Andrew Fraser, Alexei Klimenko, Nicolas Hengartner, Chris Morris, Christopher Orum, and Michael Sossong. Statistical tomographic reconstruction for cosmic ray muon tomography. Submitted, 2006.