

Article

ℓ^1 Major Component Detection and Analysis (ℓ^1 MCDA): Foundations in Two Dimensions

Ye Tian ^{1,2,*}, Qingwei Jin ^{1,3}, John E. Lavery ^{1,4} and Shu-Cherng Fang ¹

¹ Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, NC 27695-7906, USA; E-Mail: qingweijin@gmail.com (Q.J.); john.e.lavery4.civ@mail.mil (J.E.L.); fang@ncsu.edu (S.-C.F.)

² School of Business Administration, Southwestern University of Finance and Economics, Chengdu, 610074, China

³ Department of Management Science and Engineering, Zhejiang University, Hangzhou, 310058, China

⁴ Mathematical Sciences Division and Computing Sciences Division, Army Research Office, Army Research Laboratory, P.O. Box 12211, Research Triangle Park, NC 27709-2211, USA

* Author to whom correspondence should be addressed; E-Mail: tianye7272@gmail.com; Tel.: +028-87092184; Fax: +028-87092768.

Received: 5 October 2012; in revised form: 3 January 2013 / Accepted: 7 January 2013 /

Published: 17 January 2013

Abstract: Principal Component Analysis (PCA) is widely used for identifying the major components of statistically distributed point clouds. Robust versions of PCA, often based in part on the ℓ^1 norm (rather than the ℓ^2 norm), are increasingly used, especially for point clouds with many outliers. Neither standard PCA nor robust PCAs can provide, without additional assumptions, reliable information for outlier-rich point clouds and for distributions with several main directions (spokes). We carry out a fundamental and complete reformulation of the PCA approach in a framework based exclusively on the ℓ^1 norm and heavy-tailed distributions. The ℓ^1 Major Component Detection and Analysis (ℓ^1 MCDA) that we propose can determine the main directions and the radial extent of 2D data from single or multiple superimposed Gaussian or heavy-tailed distributions without and with patterned artificial outliers (clutter). In nearly all cases in the computational results, 2D ℓ^1 MCDA has accuracy superior to that of standard PCA and of two robust PCAs, namely, the projection-pursuit method of Croux and Ruiz-Gazen and the ℓ^1 factorization method of Ke and Kanade. (Standard PCA is, of course, superior to ℓ^1 MCDA for Gaussian-distributed point clouds.) The computing time of ℓ^1 MCDA is competitive with the computing times of the two robust PCAs.

Keywords: heavy-tailed distribution; ℓ^1 ; ℓ^2 ; major component; multivariate statistics; outliers; principal component analysis; 2D

Classification: MSC 62H25, 65D10

1. Introduction

Discerning the major components of a point cloud is of importance for finding patterns in the cloud and for compressing it. Principal Component Analysis (PCA) is a widely used successful tool to determine the direction, spread, and dimensionality of point clouds [1,2]). In its standard formulation, PCA is based on analysis in the square of the ℓ^2 norm applicable to data from distributions with finite second-order moments. This results in excellent performance when the point cloud has Gaussian structure or the structure of a distribution close to Gaussian. However, point clouds obtained under conditions other than benign laboratory conditions often contain significant numbers of outliers and the points may follow a heavy-tailed distribution that does not have finite second-order moments, which strongly limits the accuracy of standard PCA or prevents it from being applicable.

To remedy this situation, robust PCAs [3–6] and especially robust PCAs involving the ℓ^1 norm [7–10] have been investigated over the past few years. In most of the ℓ^1 reformulations of PCA that have been proposed, the ℓ^1 norm is applied only to parts of the PCA process. An analytical connection with heavy-tailed statistics is not present in any of these reformulations. Much of the previous work on ℓ^1 methods for PCA and in other areas has been carried out under the assumption of sparsity of the principal components or of the error. But the principal components and the error are often not sparse. While an assumption of sparsity is common in many areas (for example, in compressed sensing) and can lead to meaningful analytical and computational results in those areas, it restricts the set of situations that a reformulated PCA can address. Finally, neither standard PCA nor any of the reformulated robust PCAs are able to provide reliable information for distributions with two or more irregularly spaced main directions (spokes, such as from superimposing several classical distributions).

We hypothesize that ℓ^2 -based concepts such as singular values, inner products, orthogonal projection, averaging and second-order moments (variances and covariances) in the reformulated PCAs that have been investigated in the recent literature are limiting factors in the applicability of these PCAs to realistic outlier-rich point clouds that occur in geometric modeling, image analysis, object and face recognition, data mining, network analysis and many other areas. To remedy this situation, we carry out here a fundamental reformulation of the PCA approach in a framework based not just in part but in total on the ℓ^1 norm. The ℓ^1 norm is chosen because it is an appropriate norm when the data cloud has a significant number of outliers, either artificial outliers or outliers of a heavy-tailed statistical distribution. While analysis of multivariate heavy-tailed distributions is still underdeveloped, there has been progress [11–14]). Our ℓ^1 approach here is related to the approach by which L_1 splines, a new class of splines that preserve shape for highly irregular data, have been created over the past 12 years [15–19].

2. Standard PCA and Recently Developed Robust PCAs

In its standard formulation [2], PCA is designed to create uncorrelated components (“orthogonal solutions”). There are variants of PCA that create correlated components (“oblique solutions”), but we do not consider them here. Principal components are normally calculated using the singular value decomposition of the data matrix, an efficient and stable numerical procedure. Standard PCA can be summarized as follows:

1. Calculate the mean of the point cloud and subtract it out of the data.
2. Set up the matrix X of the data that result from Step 1. The rows of X are the data vectors.
3. Calculate the singular values of X , that is, the diagonal elements of the matrix Σ in the singular value decomposition $X = U\Sigma V^T$. (These singular values are the eigenvalues of the covariance matrix of the data.)
4. Order the components (the columns of the matrix V in the singular value decomposition) in descending order of the singular values.
5. Select basis vectors to be the components corresponding to the largest singular values.
6. Conduct further processing (for example, project the data onto the basis consisting of the vectors selected in Step 5).

Step 3 makes the limitations of standard PCA apparent. For the singular values to be meaningful, the covariance matrix of the continuum distribution from which the samples come needs to exist, that is, be finite. There are heavy-tailed distributions for which covariance matrices exist and others for which they do not. When there are a significant number of outliers in the data, an assertion that the covariance matrix of the continuum distribution exists can be dubious. A large number of outliers in the data is often an indication that the data come from a continuum heavy-tailed distribution for which the covariance matrix may not exist (has infinite entries). While the covariance matrix of a finite sample always exists, it is meaningful only if the covariance matrix of the underlying continuum distribution exists. In the remainder of this section, we discuss four robust variants of PCA that are currently in use.

Candès *et al.* [7] use the “nuclear norm” of a matrix in their robust PCA. The nuclear norm of a matrix P , denoted by $\|P\|_{\bullet}$, is the sum of the singular values of P . Let $\|E\|_1$ denote the sum of the absolute values of the entries of a matrix E . Let D be the data matrix, P be the matrix of principal components, and E be the error matrix. Candès *et al.* formulate the robust PCA problem as minimization of

$$\|P\|_{\bullet} + \lambda \|E\|_1 \quad \text{subject to } P + E = D \quad (1)$$

under assumptions that P is of low rank and E is sparse. Here, λ is a prescribed parameter based on the size of the data set [7]. Using an ℓ^1 norm to replace an ℓ^2 norm in a computational method is often a good procedure for robustifying the method. Minimization of expression (1) is seemingly based on the ℓ^1 norm, since both of the norms that occur in the objective-functional portion of expression (1) consist of sums of absolute values. However, the nuclear norm is not an ℓ^1 metric but only a “pseudo- ℓ^1 ” metric because the singular values on which it is based are created by an ℓ^2 -based process. Moreover, the arithmetic mean used to center the data is also an ℓ^2 , not an ℓ^1 quantity. It is our hypothesis that a procedure that avoids use of singular values, which are ℓ^2 quantities, and is based completely on the ℓ^1

metric will provide more accurate output information about realistic outlier-rich data than robust PCAs that are based on the use of singular values.

Kwak [10] proposes maximizing the ℓ^1 norm of the product of a vector and the data matrix and provides face recognition results that indicate success. This method uses inner products (a matrix or a vector times a matrix), which are ℓ^2 operations that do not exist in an ℓ^1 space. Here again, we surmise that a procedure that avoids use of all ℓ^2 quantities and is based completely on the ℓ^1 metric will have better performance.

Croux and Ruiz-Gazen [4] calculate robust estimates of the eigenvalues and eigenvectors of the covariance matrix without estimating the covariance matrix itself. Their method is based on a projection-pursuit approach developed by Li and Chen [20]. In this approach, one searches for directions for which the data, projected onto these directions, have maximal dispersion. This dispersion is measured not by the variance but by a robust scale estimator S_n . For data $\{\mathbf{x}_m\}_{m=0}^{M-1}$, the estimate of the first eigenvector is defined to be

$$v_{S_n,1} = \operatorname{argmax}_{\|a\|=1} S_n(a^T \mathbf{x}_0, a^T \mathbf{x}_1, \dots, a^T \mathbf{x}_{M-1}) \tag{2}$$

and the associated eigenvalue is

$$\lambda_{S_n,1} = S_n^2((v_{S_n,1})^T \mathbf{x}_0, (v_{S_n,1})^T \mathbf{x}_1, \dots, (v_{S_n,1})^T \mathbf{x}_{M-1}) \tag{3}$$

Subsequent eigenvectors are determined by searching in an analogous manner over subspaces orthogonal to all subspaces already found. Croux and Ruiz-Gazen [4] made the projection-pursuit process of Li and Chen more efficient by approximating on each step a full-space search for the vector a by a search over a finite set while retaining a high finite-sample breakdown point.

Finally, Ke and Kanade [8,9] seek a UV factorization of the data matrix $X = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{M-1})$ such that

$$\|X - UV^T\|_1 \tag{4}$$

is minimized. This method uses the ℓ^1 norm but also involves ℓ^2 inner products (matrix multiplication). The minimization of expression (4) is non-convex, so Ke and Kanade calculate U and V iteratively with a random initialization of U , optimizing one matrix while keeping the other one fixed, in the following manner. Given $U = U^{(k)}$, $V^{(k)}$ is defined to be

$$V^{(k)} = \operatorname{argmin}_V \|X - U^{(k)}V^T\|_1 \tag{5}$$

The next U , that is, $U^{(k+1)}$ is defined to be

$$U^{(k+1)} = \operatorname{argmin}_U \|X - UV^{(k)T}\|_1 \tag{6}$$

These two ℓ^1 minimization problems can be decomposed into independent minimization problems that Ke and Kanade solve by linear programming or, approximately, by quadratic programming.

In addition to the issues mentioned above, currently available robust PCAs are not able to meaningfully handle data from distributions with multiple, irregularly spaced main directions or “spokes,” that is, directions in which the level surfaces of the probability density function extend further out from the central point than in neighboring directions, a situation that is increasingly common for

some physical and a lot of non-physical (sociological, cognitive, human-behavior, *etc.*) data. These PCAs provide only one main direction that accounts for the largest variability in the data and additional orthogonal directions that account for the remaining variability. But these directions may provide very little information about the data. In the literature, one extension of PCA that detects orthogonal spokes using a union-of-subspaces model is available [21]. However, this extension does not involve the ℓ^1 norm and is therefore not robust when a significant number of outliers is present. It is possible that “superstructure” could be added to standard PCA and to robust PCAs to allow them to calculate multiple non-orthogonal spokes, but such generalizations of these methods are not yet available in the literature. There is nevertheless a need to develop a method that can calculate the directions and spreads of multiple major components.

In the next section, we consider how to reformulate the PCA approach in a framework based exclusively on the ℓ^1 norm and heavy-tailed distributions, without using any ℓ^2 -based concepts, in a way that allows calculation of the directions and spreads of single and of multiple major components. Our reformulation occurs in a manner that differs from previous ℓ^1 -based robust PCAs not merely in algorithmic structure but also in theory. The theory that we propose relies on the linkage between heavy-tailed distributions and the ℓ^1 norm, a linkage that is not discussed in the previous literature about robust PCAs.

3. 2D ℓ^1 Major Component Detection and Analysis

Reformulation of the PCA approach in the ℓ^1 norm involves more than adjusting individual steps in the standard PCA process that was outlined in Section 2. One must now accomplish the objective of determining the main directions and the magnitudes of the spread of the point cloud in those directions without the tools (singular values, inner products, orthogonal projection, averaging and second-order moments) of the standard ℓ^2 -based approach.

The ℓ^1 Major Component Detection and Analysis (ℓ^1 MCDA) that we propose consists of the following two steps:

1. Calculate the central point of the data and subtract it out of the data.
2. Calculate the main directions of the point cloud that results from Step 1 and the magnitudes of radial extension in those directions.

Post-processing analogous to Step 6 of standard PCA will be part of a fully developed ℓ^1 MCDA in the future but will not be discussed in this paper.

The point cloud under consideration is denoted by $\{\mathbf{x}_m\}_{m=0}^{M-1}$. The distance between points \mathbf{x} and \mathbf{y} in the data space is denoted by $d(\mathbf{x}, \mathbf{y})$. Since the ℓ^1 norm requires fewer operations than the ℓ^2 norm (see Remark 1 below), we will use the ℓ^1 norm to define the distance function in the data space for the description of the algorithm in the present section and for the computational experiments discussed in Section 4. However, ℓ^1 MCDA works with any distance function in the data space that the user wishes to choose, for example, the ℓ^2 norm. The distance function does need not to satisfy the triangle inequality but does need to allow definition of angular coordinates. All ℓ^p norms, $1 \leq p < \infty$, allow definition of angular coordinates. We do not require orthogonality of the main directions of the distribution but do allow orthogonality to be imposed based on outside information, for example, when we wish to

identify major components of a point cloud that is known to be from a distribution with orthogonal main directions or when the data are geometric data with orthogonal main directions in a Euclidean space.

Remark 1 Minimization of a “linear” ℓ^1 functional (sum of absolute values of linear components) is a linear programming problem that is generally more expensive to solve than minimization of a corresponding ℓ^2 functional (sum of squares of linear components), which is carried out by solving one linear system. Not surprisingly, therefore, the ℓ^1 MCDA that we will develop will be more expensive than standard ℓ^2 -based PCA. However, as a functional for measuring distance in the data space, the ℓ^1 norm is not an ℓ^1 minimization problem but rather simply a defined metric. As a metric in the data space, the ℓ^1 norm, which is a sum of absolute values, is computationally much cheaper than the ℓ^2 norm, which is a square root of a sum of products. Rotation in ℓ^1 -normed space consists of adding and/or subtracting quantities from the coordinates while rotation in ℓ^2 -normed space involves the more expensive operations of calculation of multiple sums of products. This situation suggests that using the ℓ^1 norm in the data space is meaningful. Even when the natural norm in the data space is the ℓ^2 norm, using the computationally cheaper ℓ^1 norm as an approximation of the ℓ^2 norm (rather than vice versa as has traditionally been the case) is a meaningful choice. The fact that ℓ^1 -based methods are more expensive than ℓ^2 -based methods in many higher-level situations does not change the fact that the ℓ^1 metric is much less expensive than the ℓ^2 metric at the lowest level of measuring distance in a data space.

3.1. ℓ^1 MCDA Step 1: Calculation of the Central Point

We define the multivariate median to be the point $\hat{\mathbf{x}}$ that minimizes

$$\sum_{m=0}^{M-1} d(\hat{\mathbf{x}}, \mathbf{x}_m) \quad (7)$$

In standard PCA, the central point is calculated by minimizing (7) with the square of the ℓ^2 norm as the distance function d . This yields the multidimensional average, which costs $O(M)$ (sequential) operations. However, many heavy-tailed distributions, including the Student t distribution with 1 degree of freedom that we will use in the computational experiments in this paper, do not have an average. When the continuum distribution from which a sample is drawn does not have an average, using the average of the data points in that sample for any purpose whatsoever is wrong. When d is the ℓ^1 norm, $\hat{\mathbf{x}}$ is the coordinate-wise median, an estimator of the central point [22], that consists of scalar medians (one for each coordinate direction) and costs $O(M)$ (sequential). In this paper, we will use the coordinate-wise median, which exists for all heavy-tailed and light-tailed distributions, as the central point of the data set.

Remark 2 For distributions with spokes that are not symmetrically positioned around a central point, the coordinate-wise median is not an appropriate central point. Other options such as the “ L_1 -median” investigated by Fritz *et al.* [23] in which the distance function d is the ℓ^2 norm (rather than the widely chosen square of the ℓ^2 norm square of the ℓ_2 norm) may be more meaningful. In the present paper, we will assume that all of the spokes of the data are symmetrically positioned around a central point, that opposite spokes have the same structure and, therefore, that the coordinate-wise median is an appropriate central point.

Once the central point is calculated, it is subtracted out of the data, resulting in a point cloud that is centered at the origin of the space. Since there is little possibility of confusion, we denote the data

set after the central point has been subtracted out by $\{\mathbf{x}_m\}_{m=0}^{M-1}$, the same notation used for the original data set.

3.2. ℓ^1 MCDA Step 2: Calculation of the Main Directions

After the data have been centered in Step 1, we need to calculate the main directions in which the distribution extends. Standard PCA prescribes that the main directions of the distribution are orthogonal to each other and that the measure of the extension in each of the orthogonal directions is determined by the covariance matrix. Such structure is consistent with a standard assumption of Gaussian or near-Gaussian distribution of the data. This structure is a leading factor in keeping the computational cost of standard PCA (calculated by singular value decomposition) low. At the same time, its rigidity is a cause of the limited applicability of standard PCA.

The central point of a symmetric univariate heavy-tailed distribution is its 50% quantile, the median of the distribution. The spread of a univariate heavy-tailed distribution around its central point is represented by the distance to other quantiles, for example, the 25% and 75% quantiles. The further away the 25% and 75% quantiles are from the central point, the more spread out (flatter, with heavier tails) the distribution is. One can calculate the 25% and 75% quantiles by calculating the median of the data between the 0% quantile and 50% quantile and the median of the data between the 50% quantile and 100% quantile, respectively. This procedure for calculating the 25% and 75% quantiles is used here because it can be generalized to higher dimensions. For symmetric distributions, one need, of course, calculate only the 25% or the 75% quantile, not both. The univariate situation provides the guideline for how we approach determining the properties of a multivariate heavy-tailed distribution. How heavy-tailed a multivariate distribution is in a given direction is estimated by the “median radius” of the data points in and near that direction.

Assume for now that, as previously stated, the distribution is symmetric around the origin, that is, “spokes” are in precisely opposite directions. Since the distribution is symmetric, we map, without loss of generality, every original data point $\mathbf{x}_m = (x_m, y_m)$ for which $x_m < 0$ or for which $x_m = 0$ and $y_m > 0$ to an origin-symmetric data point $(-x_m, -y_m)$. To avoid proliferation of notation, we denote each such data point $(-x_m, -y_m)$ also by \mathbf{x}_m . For each data point \mathbf{x}_m (whether an original data point or a data point obtained by mapping to the origin-symmetric point), define the ℓ^1 direction θ_m (analogous to an angle in ℓ^2 polar coordinates) to be the y -coordinate of the corresponding point on the ℓ^1 “unit circle” (unit diamond), that is,

$$\theta_m = \frac{y_m}{|x_m| + |y_m|} \tag{8}$$

Here, all of the coordinates x_m are nonnegative and the θ_m are in the interval $[-1, 1)$. In what follows, we will need periodic extension of the θ_m 's. Let $k = sM + m$ for some integer s and for $m, 0 \leq m \leq M - 1$. The θ_k for k outside the range $0 \leq m \leq M - 1$ are defined in a natural manner as

$$\theta_k = \theta_m + 2s \tag{9}$$

(This situation is a direct analogue of the fact that the standard ℓ^2 angle α of a point can be represented by $\alpha + 2\pi s$ for any integer s .) The original data points as well as the data points obtained in this manner can be represented in “polar” form as (θ_k, r_k) where $r_k = |x_k| + |y_k|$ is the ℓ^1 radius of the data point.

In what follows, we assume that the data points (θ_m, r_m) have been ordered so that the values of θ_m increase monotonically with m . To avoid proliferation of notation, we use the same notation (θ_m, r_m) for the sorted data that was used for the unsorted data.

Due to statistical variability, we cannot find good approximations of the directions in which the distribution spreads farthest simply by identifying the locally largest values of r_m (as a function of m). One estimates the directions in which the multidimensional distribution spreads farthest and the extent to which it spreads in these directions by calculating the local maxima with respect to θ of the median radius of the distribution. To find the directions in which the median radius is locally maximal, we have to approximate the data (θ_m, r_m) in a smooth manner and then find the directions θ in which this approximation is locally maximal. A method for finding local maxima that relies on fitting the data with a global function would be computationally feasible in 2D. However, fitting a global surface to the data would be less computationally attractive in 3D and 4D (space + time) and not at all computationally useful in n dimensions for $n > 4$. For this reason, we adopt the following locally based algorithm for calculating values of a function $r(\theta)$ that represents the median radius of the data points (θ_m, r_m) and for identifying local maxima of this function:

1. Choose a point θ_m from which to start.
2. Choose an integer q that represents the number of neighbors in each direction (index lower and higher than m) that will be included in a local domain D .
3. On the local domain $D = [\theta_{m-q}, \theta_{m+q}]$, calculate the quadratic polynomial $a_0 + a_1\theta + a_2\theta^2$ that best fits the data in the ℓ^1 norm, that is, minimizes,

$$\sum_{k=m-q}^{m+q} |a_0 + a_1\theta_k + a_2\theta_k^2 - r_k| \tag{10}$$

over all real numbers a_0, a_1 and a_2 .

4. Determine the location of the maximum of the quadratic polynomial on the local domain D . If the location of the maximum is strictly inside D , go to Step 5. If the location of the maximum is at θ_{m-q} or θ_{m+q} , call this direction a new θ_m and return to Step 3.
5. Refine the location and value of the maximum of the median radius in the following way. Calculate a quadratic polynomial on a local domain with a larger q^* around the node closest to the location of the approximate local maximum identified in Step 4. The maximum of this quadratic polynomial is the estimate of the maximum of the median radius.

The above procedure is for calculating one local maximum. To calculate all local maxima for a distribution with several spokes, one needs to assume that the spokes are distinct from each other, that is, do not overlap in a way that two closely placed spokes appear nearly like one spoke. For example, one may have information that the spokes are locally separated by angular distances greater than or equal to a known lower bound. One then chooses starting points for multiple implementations of Step 1 that cover the interval $[-1, 1)$ at distances that are slightly less than the lower bound. For applications for which local minima are important, one can calculate local minima analogously.

Remark 3 The (unusual) situation in which all points lie on one or a few radial lines does not allow use of the local fitting method described above and can be taken care of by other procedures. In this

case, the maxima of the median radius occur at the directions of the radial lines. From the clusters of points with identical θ_i , one identifies the directions of the lines and then calculates the one-dimensional median of the r_i in each cluster.

Theoretical guidance for choosing the q and q^* of Steps 2 and 5 is not yet available but it is clear that q and q^* need to be chosen based on the structure of the distribution and the properties of the outliers in the data. There will certainly be a trade-off between accuracy and computational efficiency. With more noise and outliers, one will need to use larger local domains (more neighbors, that is, larger q and/or q^*) to retain sufficient accuracy.

4. Computational Experiments

In this section, we present comparisons of 2D ℓ^1 MCDA with standard PCA, Croux and Ruiz-Gazen's method and Ke and Kanade's method. Of all the robust PCAs that have been developed, only Ke and Kanade's method [8,9], uses ℓ^1 as its main basis. For this reason, comparison of our ℓ^1 MCDA, which is "fully ℓ^1 ", with Ke and Kanade's method is important. For further contextual awareness, comparison with another widely used robust PCA, for example, Croux and Ruiz-Gazen's projection-pursuit method [4] is equally important.

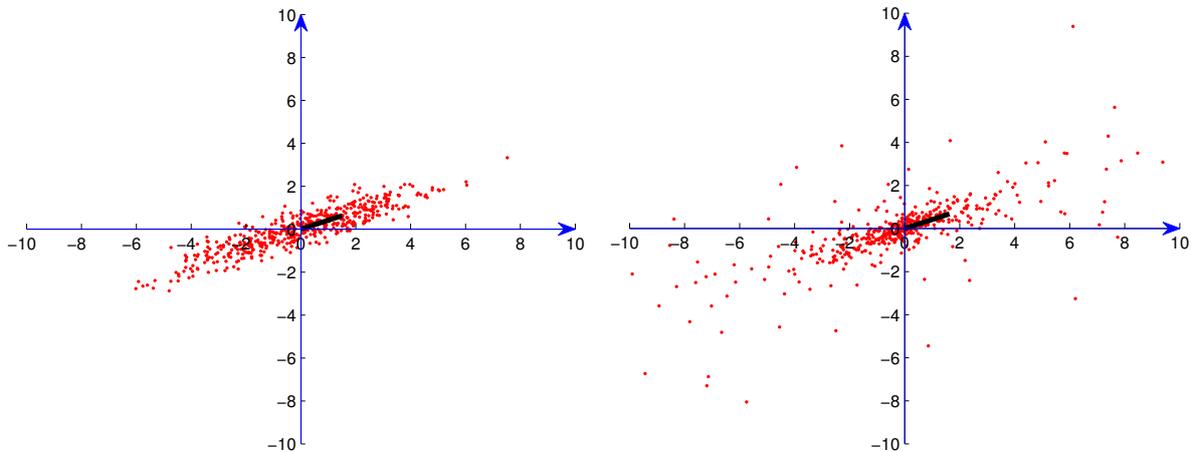
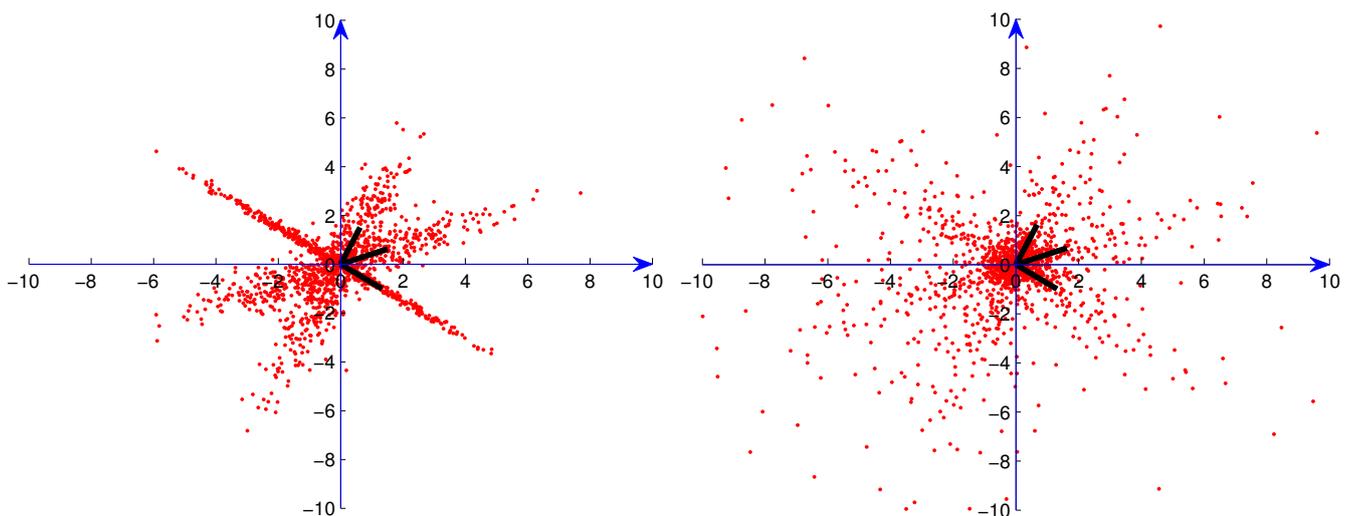
Eight types of distributions were used for the computational experiments:

- Bivariate Gaussian without and with additional artificial outliers
- Bivariate Student t with 1 degree of freedom without and with additional artificial outliers
- Three superimposed bivariate Gaussians without and with additional artificial outliers
- Three superimposed bivariate Student t with 1 degree of freedom without and with additional artificial outliers

Bivariate Student t distributions with 1 degree of freedom are heavy-tailed distributions with particularly heavy tails. These distributions were chosen for the computational experiments because they represent a significant computational challenge for standard PCA, the robust PCAs and ℓ^1 MCDA.

All computational results were generated by MATLAB R2009b on a sequential 2.50 GHz computer with 1GB memory. The quadratic functions in Steps 3 and 5 of the local ℓ^1 fitting algorithm described in Subsection 3.2 were calculated by the MATLAB *linprog* module. The q and q^* of Steps 2 and 5 of the algorithm for calculating the directions and values of the local maxima of the median radius were chosen to be 5 and 12, respectively.

We generated samples from distributions with median ℓ^1 radius ρ in the ℓ^1 direction α for the following ρ and α . For the one-main-direction situation, we used samples from bivariate distributions with $\{\rho, \alpha\} = \{2.1291, 0.2929\}$. Distributions with three main directions were generated by superimposing three distributions with $\{\rho, \alpha\} = \{2.3340, -0.4308\}$, $\{2.1291, 0.2929\}$ and $\{2.1291, 0.7071\}$. Samples from the bivariate Student t distributions were generated by the Matlab *mvtrnd* module using the correlation matrix $A = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$. These samples were rotated (in ℓ^2) to the ℓ^1 directions mentioned above. In the figures described below in which we illustrate these distributions, we indicate the points of these samples by red dots. Directions and magnitudes of the maxima of the median radius are indicated by bars extending out from the origin.

Figure 1. Sample from one Gaussian distribution without artificial outliers.**Figure 2.** Sample from three superimposed Gaussian distributions without artificial outliers.

In the computational experiments, we used data sets consisting of 50, 150 and 500 points. In Figures 1–4, we present examples of the data sets with 500 points (red dots) for one Gaussian distribution, one Student t distribution, three superimposed Gaussian distributions and three superimposed Student t distributions, respectively. In order to exhibit the major components clearly, we show in these figures data points only in $[-10, 10] \times [-10, 10]$. For Student t distributions, there are still many points outside $[-10, 10] \times [-10, 10]$. We also generated data sets consisting of data from the distributions described above along with 5%, 10% and 20% patterned artificial outliers that represent clutter. For the one-distribution situation, artificial outliers were set up using a uniform statistical distribution on the ℓ^1 diamond with radius 1000 in the ℓ^1 -direction window $[-0.9, -0.5]$. For the three-superimposed-distribution situation, artificial outliers were set up using uniform statistical distributions on the ℓ^1 diamond with radius 1000 in the three different ℓ^1 -direction windows $[-0.9, -0.8]$, $[-0.3, -0.2]$ and $[0.4, 0.5]$. In Figs. 5, 6, 7 and 8, we present examples of 500-point data sets with 10% artificial outliers (depicted by blue dots) for one Gaussian distribution, one Student t distribution,

three superimposed Gaussian distributions and three superimposed Student t distributions (points from distributions depicted by red dots), respectively.

Figure 3. Sample from one Gaussian distribution with 10% artificial outliers.

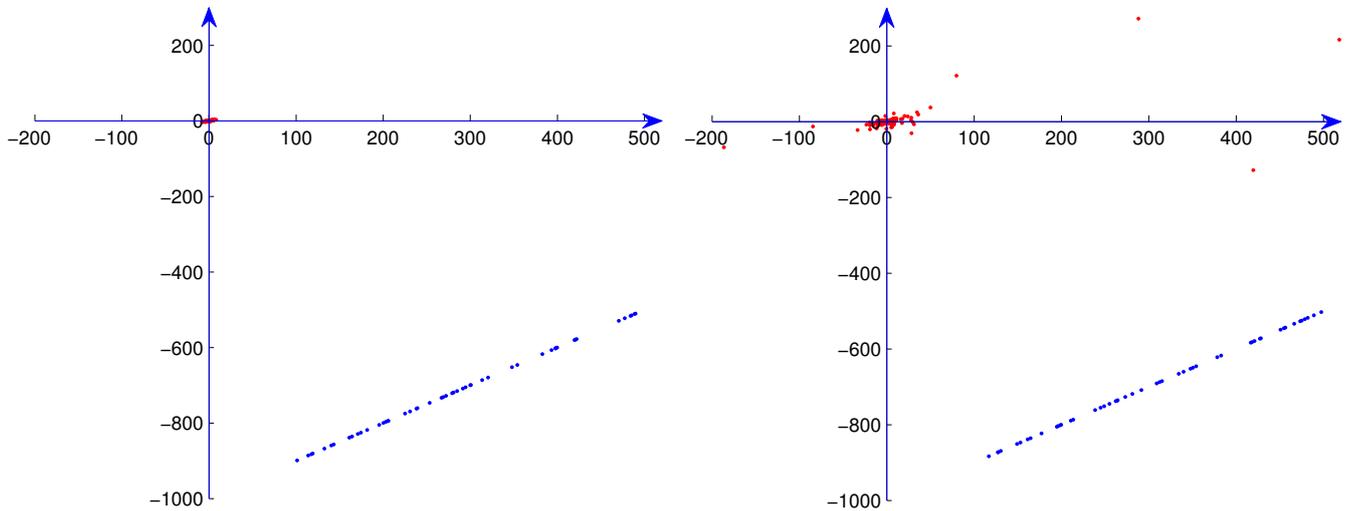
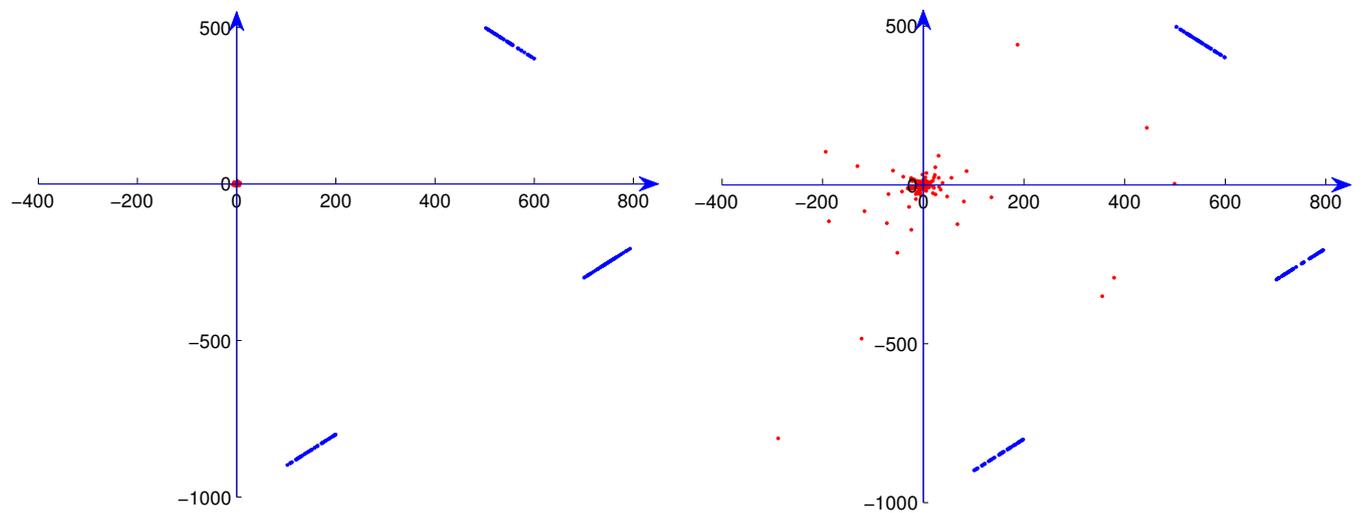


Figure 4. Sample from three superimposed Gaussian distributions with 10% artificial outliers.



For each type of data, we carried out 100 computational experiments, each time with a new sample from the statistical distribution(s), including the uniform distributions that generated the outliers. To measure the accuracy of the results, we calculated the average and standard deviation (over 100 computational experiments) of the error of each main ℓ^1 direction and the average and standard deviation of the error of the median radius in that direction vs. the theoretical values of the direction of maximum spread and the median radius in that direction of the continuum distribution. In Tables 1–5, we present computational results for the sets of 500 points. Computational results for 50 and 150 points were analogous to those for 500 points.

Table 1. Average error (av. error) and standard deviation of the error (std. dev.) of ℓ^1 direction θ for one Gaussian distribution.

	Standard PCA	Croux+Ruiz-Gazen	Ke+Kanade	ℓ^1 MCDA
av. error	0.00005	-0.0034	-0.0063	-0.0081
std. dev.	0.0051	0.0563	0.0084	0.0105
av. error	-0.0053	0.0219	—	0.0349
std. dev.	0.0901	0.1733	—	0.1683

Table 2. Average error (av. error) and standard deviation of the error (std. dev.) of ℓ^1 direction θ for one Gaussian distribution with 10% artificial outliers.

	Standard PCA	Croux+Ruiz-Gazen	Ke+Kanade	ℓ^1 MCDA
av. error	1.0046	-0.0168	0.9148	-0.0105
std. dev.	0.0184	0.0837	0.0243	0.0114
av. error	261.9384	0.3185	—	0.1138
std. dev.	3.4384	0.1482	—	0.2028

Table 3. Average error (av. error) and standard deviation of the error (std. dev.) of ℓ^1 direction θ for one Student t distribution.

	Standard PCA	Croux+Ruiz-Gazen	Ke+Kanade	ℓ^1 MCDA
av. error	0.0353	0.0614	0.0474	0.0163
std. dev.	0.1983	0.0743	0.1738	0.01717
av. error	147.2380	-0.7139	—	0.1438
std. dev.	258.1932	0.1692	—	0.2302

Table 4. Average error (av. error) and standard deviation of the error (std. dev.) of ℓ^1 direction θ for one Student t distribution with 10% artificial outliers.

	Standard PCA	Croux+Ruiz-Gazen	Ke+Kanade	ℓ^1 MCDA
av. error	1.0023	0.0624	-0.8274	-0.0213
std. dev.	0.2683	0.1744	0.2038	0.0234
av. error	385.4844	-0.2839	—	0.2301
std. dev.	212.3327	0.2732	—	0.2289

Table 5. Average error (av. error) and standard deviation of the error (std. dev.) of ℓ^1 direction θ calculated by ℓ^1 MCDA for three superimposed distributions without and with 10% artificial outliers.

Distribution	av. error	std. dev.
3 Gaussians	-0.0091	0.0103
	0.0124	0.0201
	0.0111	0.0103
3 Gaussians with 10% outliers	-0.0093	0.0221
	-0.0084	0.0252
	0.0124	0.0110
3 Student t	-0.0092	0.0224
	0.0130	0.0202
	-0.0110	0.0193
3 Student t with 10% outliers	0.0143	0.0224
	-0.0159	0.0252
	0.0130	0.0243
3 Gaussians	0.0832	0.1593
	0.1291	0.1788
	0.1402	0.1891
3 Gaussians with 10% outliers	0.1382	0.1632
	0.1389	0.1738
	0.1537	0.1838
3 Student t	0.1839	0.2582
	0.1783	0.2633
	0.1478	0.2537
3 Student t with 10% outliers	0.1987	0.2638
	0.1733	0.2837
	0.2018	0.2738

Remark 4 Since ℓ^1 MCDA is an ℓ^1 method, one may ask why the accuracy of the results is measured using averages and standard deviations rather than, for example, ℓ^1 measures such as medians and other quantiles. The statistical distributions of the directions and median radii that are calculated by ℓ^1 MCDA are zero-tailed and (apparently) light-tailed, respectively, which indicates that averages and standard deviations are more appropriate measures than quantiles.

Remark 5 No results for Ke and Kanade’s method are provided in Tables 1–4 because Ke and Kanade’s method does not yield radius information.

The results in Table 1 indicate that, as theoretically predicted, standard PCA performs better than any of the other three methods for data from one single Gaussian distribution. The results in Tables 2–4 indicate that, as expected, standard PCA does not handle data with artificial outliers and/or from heavy-tailed distributions well. The results in Tables 2–4 show, consistent with theoretical and

computational evidence available in the previous literature, the advantages of Croux and Ruiz-Gazen’s projection-pursuit method and of Ke and Kanade’s ℓ^1 factorization method over standard PCA. These results also show in nearly all cases a marked advantage of ℓ^1 MCDA over Croux and Ruiz-Gazen’s projection-pursuit and Ke and Kanade’s ℓ^1 factorization. For the multiple superimposed distributions considered in Table 5, standard PCA, Croux and Ruiz-Gazen’s projection-pursuit and Ke and Kanade’s ℓ^1 factorization each provide only one main direction for the superimposed distributions and do not yield any meaningful information about the individual main directions. For this reason, no results for these three methods are presented in Table 5. It is worth noting in Table 5 that the accuracy of ℓ^1 MCDA for three superimposed distributions without and with artificial outliers is just as good as the accuracy of ℓ^1 MCDA for single distributions without and with artificial outliers.

As the computing times reported in Table 6 indicate, 2D ℓ^1 MCDA in its current implementation costs 30 to 40 times as much as standard PCA, roughly 3 times as much as Ke and Kanade’s factorization and 30% to 40% less than Croux and Ruiz-Gazen’s projection-pursuit (all methods sequentially implemented). The wide applicability of ℓ^1 MCDA in comparison with Ke and Kanade’s factorization method justifies an increase of computing time by a factor of 3. Moreover, the computing time of ℓ^1 MCDA is likely to decrease as the method is further investigated and more efficient implementations are designed.

Table 6. Sequential computing times for generating the results in Tables 1–5 by the four methods.

Data of	Standard PCA	Croux+Ruiz-Gazen	Ke+Kanade	ℓ^1 MCDA
Table 1	3.398ms	177.832ms	32.921ms	107.382ms
Table 2	4.411ms	198.833ms	33.477ms	116.504ms
Table 3	3.667ms	197.221ms	48.133ms	131.338ms
Table 4	5.277ms	210.442ms	55.672ms	147.185ms
				225.348ms
Table 5				248.392ms
				299.392ms
				335.429ms

Computational results for 5% artificial outliers were analogous to the results for 10% artificial outliers. Computational results for ℓ^1 MCDA with 20% artificial outliers in the data were not as accurate as those for 10% outliers. A quantitative description of the robustness of ℓ^1 MCDA in terms of breakdown point (proportion of outliers beyond which the method produces errors that are arbitrarily large) or influence function (how the result changes when one point is changed) will depend not only on the radii and angular coordinates of the points that are changed but also on clustering patterns of these points in relation to the other points. This task is beyond the scope of this paper but will be an objective of future research.

The results in Tables 1–5 indicate that, for the data considered here, ℓ^1 MCDA always outperforms standard PCA in accuracy except when the data come from one single Gaussian distribution and that ℓ^1 MCDA outperforms two robust PCAs in accuracy in nearly all cases in accuracy in nearly all the cases

presented here. These computational results are consistent with the theoretically known fact that standard PCA is optimal for one single Gaussian distribution. Interestingly, however, the results in Table 1 indicate that ℓ^1 MCDA performs quite well—albeit sub-optimally—for one single Gaussian distribution. Thus, there is no major disadvantage in using ℓ^1 MCDA as a default PCA (instead of standard PCA), since it performs well both for the cases when standard PCA is known to be optimal and, as we have seen, for the many other cases when standard PCA and robust PCAs perform less well, poorly or not at all.

5. Conclusions and Future Work

The assumptions underlying ℓ^1 MCDA are less restrictive and more practical than those underlying standard PCA and currently available robust PCAs. The 2D ℓ^1 MCDA that we have developed differs from standard PCA and all previously proposed robust PCAs in that it (1) allows use of a wide variety of distance functions in the data space (while noting the advantages of using the ℓ^1 norm to define the distance function); (2) replaces all (not just some) of the ℓ^2 -based procedures and concepts in standard PCA with ℓ^1 -based procedures and concepts; (3) has a theoretical foundation in heavy-tailed statistics but works well for data from both heavy-tailed and light-tailed distributions; (4) is applicable for data that need not have (but can have) mutually orthogonal main directions, can have multiple spokes and can contain patterned artificial outliers (clutter) and (5) does not require assumption of sparsity of the principal components or of the error. Most of the robust PCAs (with the exception of Ke and Kanade's) that have previously been proposed in the literature involve use of the ℓ^1 norm not at all or only to a limited extent and continue to rely on ℓ^2 -based items including singular values, inner products, orthogonal projection, averaging and second moments (variances, covariances). The ℓ^1 MCDA that we propose comes exclusively from a unified theoretical framework based on the ℓ^1 norm. The computational results presented in Section 4 show that the ℓ^1 MCDA proposed here significantly outperforms not only the standard PCA but also two robust PCAs in terms of accuracy.

This ℓ^1 MCDA is a foundation for a new, robust procedure that can be used for identification of dimensionality, identification of structure (including nonconventional spoke structure) and data compression in \mathbb{R}^n , $n \geq 3$, a topic on which the authors of this paper are currently working. In designing ℓ^1 MCDA for higher dimensions, the guiding principles will continue to be direct connection with heavy-tailed statistics and exclusive reliance on ℓ^1 operations. The higher-dimensional versions of Steps 1 and 2 of ℓ^1 MCDA are feasible as long as appropriate higher-dimensional angular coordinates can be defined. The reader may question whether the " ℓ^1 polar coordinates" that are used in 2D can be extended to n dimensions. Indeed they can in the following manner. In direct analogy to the definition of the polar coordinate θ_m in (8) in two dimensions, one defines angular coordinates one defines $n - 1$ angular coordinates of an n -dimensional data point $(x_m^{(1)}, x_m^{(2)}, \dots, x_m^{(n)})$ to be the quotients with the components of the point as numerators and the ℓ^1 radius $\sum_{j=1}^n |x_m^{(j)}|$ of the point as denominators. In passing we note that these higher-dimensional ℓ^1 angular coordinates are computationally cheaper than standard ℓ^2 hyperspherical angular coordinates, which require calculation of square roots of sums of squares.

For samples of M vectors in \mathbb{R}^n , $M > n$, the cost of classical PCA is $O(Mn^2)$. The costs of Croux and Ruiz-Gazen's method and of Ke and Kanade's method are not specified in the literature but

apparently scale linearly with respect to sample size M . We hypothesize that the extension of ℓ^1 MCDA to higher dimensions will have a sequential cost of $O(Mn)$ and will be comparable with or lower than the cost of competing robust PCAs.

ℓ^1 MCDA is expected to be a robust tool in terrain modeling, geometric modeling, image analysis, information mining, face/object recognition and general pattern recognition. As suggested by the computational results presented in the present paper, it is expected that ℓ^1 MCDA will be particularly useful for pattern recognition under patterned clutter. For example, ℓ^1 MCDA will be useful for identification of objects behind occlusions because it handles the occlusions as outliers and does not require a separate step of segmenting out the occlusions. This capability will provide a basis for going directly from point cloud to robust semantic labeling.

Acknowledgements

The authors wish to thank two anonymous reviewers for their insightful comments and questions, which led to improvements in the paper. This work was generously supported by the DARPA Adaptive Execution Office (Todd Hughes) through US Army Research Office Grant # W911NF-04-D-0003, by the North Carolina State University Edward P. Fitts Fellowship and by US NSF Grant # DMI-0553310. It is the policy of the Army Research Office that university personnel do not need to do joint work with ARO personnel in order to receive grants from the Army Research Office.

References

1. Gorban, A.; Kegl, B.; Wunsch, D.; Zinovyev, A. *Principal Manifolds for Data Visualisation and Dimension Reduction*; Springer: New York, NY, USA, 2007.
2. Jolliffe, I.T. *Principal Component Analysis*, 2nd ed.; Springer: New York, NY, USA, 2002.
3. Chen, T.; Martin, E.; Montague, G. Robust probabilistic PCA with missing data and contribution analysis for outlier detection. *Comput. Stat. Data Anal.* **2009**, *53*, 3706–3716.
4. Croux, C.; Ruiz-Gazen, A. High breakdown estimators for principal components: The projection-pursuit approach revisited. *J. Multivar. Anal.* **2005**, *95*, 206–226.
5. Hubert, M.; Rousseeuw, P.; Verdonck, T. Robust PCA for skewed data and its outlier map. *Comput. Stat. Data Anal.* **2009**, *53*, 2264–2274.
6. Serneels, S.; Verdonck, T. Principal component analysis for data containing outliers and missing elements. *Comput. Stat. Data Anal.* **2008**, *52*, 1712–1727.
7. Candès, E.J.; Li, X.; Ma, Y.; Wright, J. *Robust Principal Component Analysis*; Technical Report No. 13.; Department of Statistics, Stanford University, Stanford, CA, USA, 2009.
8. Ke, Q.; Kanade, T. *Robust Subspace Computation Using L_1 Norm*; Technical Report CMU-CS-03-172. Carnegie Mellon University: Pittsburgh, PA, USA, 2003; Available online: <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.63.4605> (accessed on 14 January 2013).
9. Ke, Q.; Kanade, T. Robust L_1 Norm Factorization in the Presence of Outliers and Missing Data by Alternative Convex Programming. In *Proceedings of the IEEE Conference Computer Vision Pattern Recognition*, San Diego, California, USA, 20–25 June 2005; Volume 1, pp. 739–746.

10. Kwak, N. Principal component analysis based on L_1 -norm maximization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1672–1680.
11. Nolan, J.P. *Multivariate Stable Distributions: Approximation, Estimation, Simulation and Identification*; Adler, R.J., Feldman, R.E., Taqqu, M.S., Ed.; A Practical Guide to Heavy Tails, Birkhauser, Cambridge, UK, 1998; pp. 509–525.
12. Nolan, J.P.; Panorska, A.K. Data analysis for heavy tailed multivariate samples. *Stoch. Models* **1997**, *13*, 687–702.
13. Resnick, S.I. On the foundations of multivariate heavy-tail analysis. *J. Appl. Probab.* **2004**, *41*, 191–212.
14. Resnick, S.I. *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling*; Springer-Verlag: Berlin, Germany, 2007.
15. Auquiart, P.; Gibaru, O.; Nyiri, E. Fast L_1 - C^k polynomial spline interpolation algorithm with shape-preserving properties. *Comput. Aided Geom. Des.* **2011**, *28*, 65–74.
16. Cheng, H.; Fang, S.-C.; Lavery, J.E. Univariate cubic L_1 splines: A geometric programming approach. *Math. Methods Oper. Res.* **2002**, *56*, 197–229.
17. Jin, Q.; Lavery, J.E.; Fang, S.-C. Univariate cubic L_1 interpolating splines: Analytical results for linearity, convexity and oscillation on 5-point windows. *Algorithms* **2010**, *3*, 276–293.
18. Lavery, J.E. Univariate cubic L_p splines and shape-preserving, multiscale interpolation by univariate cubic L_1 splines. *Comput. Aided Geom. Des.* **2000**, *17*, 319–336.
19. Yu, L.; Jin, Q.; Lavery, J.E.; Fang, S.-C. Univariate cubic L_1 interpolating splines: Spline functional, window size and analysis-based algorithm. *Algorithms* **2010**, *3*, 311–328. Available online: <http://www.mdpi.com/journal/algorithms> (accessed on 14 January 2013).
20. Li, G.; Chen, Z. Projection-pursuit approach to robust dispersion matrices and principal components: Primary theory and Monte Carlo. *J. Am. Stat. Assoc.* **1985**, *80*, 759–766.
21. Ma, Y.; Yang, A.Y.; Derksen, H.; Fossum, R. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Rev.* **2008**, *50*, 413–458.
22. Small, C.G. A survey of multidimensional medians. *Int. Stat. Rev.* **1990**, *58*, 263–277.
23. Fritz, H.; Filzmoser, P.; Croux, C. A comparison of algorithms for the multivariate L_1 -median. *Comput. Stat.* **2012**, *27*, 393–410.
24. Bulatov, D.; Lavery, J.E. Comparison of Reconstruction and Texturing of 3D Urban Terrain by L_1 Splines, Conventional Splines and Alpha Shapes. In *Proceedings of the Fourth International Conference Computer Vision Theory and Applications. VISAPP 2*, Lisbon, Portugal, 5–8 February 2009; pp. 403–409.