2011

Labeled Sampling Consensus A Novel Algorithm For Robustly Fitting Multiple Structures Using Compressed Sampling

Carl J. Messina
University of Central Florida

LABELED SAMPLING CONSENSUS: A NOVEL ALGORITHM FOR ROBUSTLY FITTING MULTIPLE STRUCTURES USING COMPRESSED SAMPLING

by

CARL J. MESSINA
B.S. University of Central Florida 2002

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2011

Major Professor:
Hassan Foroosh

ABSTRACT

The ability to robustly fit structures in datasets that contain outliers is a very important task in Image Processing, Pattern Recognition and Computer Vision. Random Sampling Consensus or RANSAC is a very popular method for this task, due to its ability to handle over 50% outliers. The problem with RANSAC is that it is only capable of finding a single structure. Therefore, if a dataset contains multiple structures, they must be found sequentially by finding the best fit, removing the points, and repeating the process. However, removing incorrect points from the dataset could prove disastrous. This thesis offers a novel approach to sampling consensus that extends its ability to discover multiple structures in a single iteration through the dataset. The process introduced is an unsupervised method, requiring no previous knowledge to the distribution of the input data. It uniquely assigns labels to different instances of similar structures. The algorithm is thus called Labeled Sampling Consensus or L-SAC. These unique instances will tend to cluster around one another allowing the individual structures to be extracted using simple clustering techniques. Since divisions instead of modes are analyzed, only a single instance of a structure need be recovered. This ability of L-SAC allows a novel sampling procedure to be presented “compressing” the required samples needed compared to traditional sampling schemes while ensuring all structures have been found. L-SAC is a flexible framework that can be applied to many problem domains.

For Mom, Dad, and Laura

ACKNOWLEDGMENTS

I would like to thank Dr. Hassan Foroosh for encouraging me to attempt a thesis and providing me with the guidance and direction needed to complete it. I also thank Dr. Charles Hughes and Dr. Marshall Tappen for agreeing to and taking the time to serve on my committee. Lastly I thank my wife, Laura, for spending many weekends alone while I undertook this endeavor.

Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

One very important task in engineering and computer science is to fit a measured data set to its ideal structure. For example, in the ideal case, all points on a line exactly follow the structure $y = mx + b$. However, when taking measurements in the real case, the points associated with a particular line will not exactly fit this structure due to introduced error. All measuring devices such as sensors and cameras have inherent limitations which introduce errors into the ideal system. These errors may be due to the discrete sampling and quantization of a signal in an ADC, or the rounding and truncation that occurs in a CPU. The ambient lighting may cause a camera's CCD to operate outside of its dynamic range, distorting its color accuracy. These errors are system noise. Whatever the case may be, it is important to find an accurate estimate of the ideal structure in this noisy data.

Noise may not be the only source of error in the system either. It is often necessary to extract features or a sub-set from the original dataset. The algorithms used for this often introduce errors themselves. Incorrect samples not associated with a structure may be gathered, or points may be matched incorrectly in a stereo vision system. These are outliers since they do not match the features extracted from the sampled set. Performing operations such as Least-Squares on a set containing outliers is not desired because the outliers merge with the valid data and skew the results. To overcome this problem, several methods for examining noisy data, and data contaminated with outliers have been developed. One popular method for robust regression is Least Median of Squares or LMedS [11], which scores structures

based on the median distances to all points. Its advantage is that no prior knowledge to the distribution is needed, however it will fail if the proportion of outliers is greater than 50%. If a dataset contains outliers greater than 50%, then a popular method in Computer Vision is to use Random Sample Consensus or RANSAC.

RANSAC introduced by Fischler and Bolles [5] in 1981 is a very popular algorithm for robustly fitting data in a noisy environment often tolerating more than 50% outliers. RANSAC proceeds by drawing a minimum number of points needed to fit the structure, measuring the number of inliers to the formed structure, and scoring the fit based on the number of inliers. A consensus set it built and updated every time a newly formed structure contains more inliers then the consensus set before it. The number of random samples required to find the best fit up to a given probability is determined based on the estimated number of outliers, and the minimum number of points required to represent a structure. This estimate for inliers can be adapted based on the consensus set to improve the algorithms performance. Once the required number of samples has been reached, the consensus set is used to fit the data. RANSAC is very popular because it can be applied to many problem sets such as fitting lines, finding planes and homographies, finding the Fundamental Matrix, and motion estimation just to name a few. It is also relatively simple to implement and use.

1.1 Motivation

RANSAC is limited to an either/or situation due to its reliance on a consensus set that is updated whenever a structure with a higher score is found. Therefore RANSAC is limited to finding only a single structure in the dataset presented. Several papers in the literature deal with extending RANSAC to single structures such as [16] and [18]. Only dealing with single structures is a severe limitation as a dataset may contain several structures, and an attempt to find them sequentially by removing structures after detection may cause serious problems if incorrect points are removed. Therefore, a novel sampling consensus algorithm, capable of handling multiple structures, has been developed to overcome this limitation of RANSAC.

This thesis introduces a novel method for finding multiple structures in a dataset. Detecting multiple structures in a dataset introduces several challenges that need to be overcome. The first challenge is that of dimensionality. A RANSAC-like algorithm depends upon a minimum sample set, that can instantiate a geometry, typically from a relatively simple mathematical relationship. Detecting a line is a $\mathbb{R}^2$ problem and homographies are $\mathbb{R}^4$. What about more complex geometrical structures such as human faces? As the geometry becomes more complex the dimensionality may increase exponentially and a closed form solution may not exist. Therefore for a RANSAC-like algorithm to work for all but trivial cases, it must handle $\mathbb{R}^n$ without a similar scaling in computational complexity.

To be RANSAC-like, the approach must be robust to outliers. This problem is two-fold when dealing with multiple structures. Gross outliers will exist in the dataset; however points that are inliers to one structure may be outliers to another structure. Therefore a dataset

containing more than one structure has both gross outliers and pseudo-outliers. These pseudo-outliers must be handled properly as they affect the consensus sets differently.

Another difficulty in handling multiple structures is determining if all structures have been found during the random sampling process. The trivial case is to know beforehand how many structures exist, but what if this information is not known? Pseudo-outliers also mean that a refinement to an estimate of the inlier/outlier ratio cannot be obtained during the sampling process as can be for the single case of RANSAC. Therefore the only way to rely on an inlier/outlier ratio for computing the number of iterations is to have *a priori* knowledge of the input data.

Finally, finding all structures in a dataset can dramatically affect the sampling procedure. Three parameters, the number of structures, dimensionality of the minimum sample set, and size of the dataset all affect how many samples must be taken to ensure with a desired probability ρ, that at least m outlier free minimum sample sets have been drawn. If n structures exist, then n different outlier free consensus sets must be drawn.

These issues must successfully be addressed for robust recovery of multiple structures. The following section describes the current state of the art.

1.2 Literature Review

This section presents the current methods existing in the literature to handle a dataset containing several structures. The approach of several specific algorithms are discussed and analyzed to determine how this thesis uniquely contributes to the current state of the art.

Mean shift and Randomized Hough Transform are popular clustering techniques that have been around for several years. Mean shift, made popular in the Computer Vision community by [2], is a technique for finding modes in probability density functions. The mean shift vector locally points in the direction of the maximum increase in the density, and tends to converge at the modes [23]. Mean shift proves difficult in multi-modal data as the choice of bandwidth greatly affects performance. If the bandwidth is too tight, it may be sensitive to local maxima, while correct peaks may be missed if the bandwidth is too wide [23].

Randomized Hough Transform (RHT) [20] differs from mean shift in that it builds histograms in the parameter space. The computational complexity associated with the traditional Hough Transform [3] is compensated by building model hypotheses from random samples. Peaks in parameter space will tend to build around a structure in the dataset. For multiple structures, the histogram will become multi-modal. RHT still suffers from the computational complexity problem and limited accuracy, while the choice of parameter space critically affects performance [15].

These two methods were not specifically designed to handle multi-structure data but were adapted, due to their mode finding abilities. They are not therefore inherently robust in

this type of environment. Over the last few years, techniques specifically designed around multi-structure environments have been developed. These methods are based on the sampling consensus procedure of RANSAC and are more suited to this problem domain.

In 2005, Zulliani*, et. al.* [25] published the multiRANSAC algorithm. The premise of this algorithm is that M structures are known to exist in the dataset $\mathrm{D} = \{x_1, x_2, \dots, x_N\}$. The set $\mathrm{D}_I = \{N_1, N_2, \dots, N_M\}$ is the set of inliers to structures $1{:}\,M$. A manifold $Z_m(\theta)$ of all points associated with the parameter vector $\theta \in \mathbb{R}^{k_m}$ is defined for structures $1 \leq m \leq M$, and this manifold has dimension k_m. The minimum elements required to instantiate a structure is the subset $s_m \subseteq \mathrm{D}$ of $k_m$also called the minimum sample set (MSS). The consensus set (CS) which represents the best fit for structure m, is found by minimizing the error ε produced by an applicable distance function

$$\varepsilon\big(x, Z_m(\theta_m)\big) = \min d\big(x, Z_m(\theta)\big). \tag{1.1}$$

The subset of points in D_I whose distance to $Z_m(\theta_m)$ is such that, $\varepsilon < \delta$, where δ is a user defined threshold, represents the CS $S(\theta_m)$.

The multiRANSAC algorithm searches the dataset by instantiating the MSS of structure m, finding the CS, then removing the inliers from D. It does this M times. The probability of finding M outlier free structures is given by,

$$q = \frac{\binom{N_{I,1}}{k_1}\binom{N_{I,2}}{k_2}\cdots\binom{N_{I,M}}{k_M}}{\binom{N}{k_1}\binom{N-N_{I,1}}{k_2}\cdots\binom{N-\sum_{m=1}^{M-1}N_{I,M}}{k_m}}. \tag{1.2}$$

Since the number of inliers for each CS is not known *a priori*, q cannot be found. However, the fact that $S(\theta) \leq \left|N_{I,m}\right|$ shows that $q(\theta) \leq q$ and conversely $1 - q \leq 1 - q(\theta)$, where $\theta = \text{cat}(\theta_1, \theta_2, \dots, \theta_M)$ and the consensus sets are sorted such that $N_{I,1} \leq \cdots \leq N_{I,M}$. With this knowledge, a stopping threshold can be calculated. The probability of not selecting M outlier free consensus sets is given by $1 - q(\theta)$. If the number of iterations is h, then the probability of not selecting M outlier free consensus sets is given by $(1 - q(\theta))^h$. As h increases, the probability goes to zero. The required number of iterations h, can be determined to a specified probability ρ by $(1 - q(\theta))^h \leq \rho$. The number of iterations is then,

$$T = \frac{\log(\rho)}{\log(1 - q(\theta))}. \tag{1.3}$$

This threshold T, is updated throughout the selection process.

MultiRANSAC is a greedy algorithm as it attempts to increase the CS at iteration h with the CS from iteration $h - 1$. If the set S with maximum cardinality is disjoint from any set in $S^{(h)}(\theta)$, then $S^{(h)}(\theta) = S^{(h)}(\theta) \cup S$.

There are two main drawbacks to using multiRANSAC. The first is that it is a supervised approach as the number of structures in the dataset is known *a priori*. The second is that multiRANSAC tends to fail if structures intersect each other as noted in [15]. This is primarily due to the greedy nature of merging consensus sets as it implicitly assumes non-intersecting structures.

Zhang *et. al.* proposed a novel approach to the RANSAC hypothesis testing approach. "Instead of considering the residuals of all the data points per hypothesis, we propose to

analyze the distribution with respect to all the hypotheses for each data point [23].” The idea is to build a histogram of the distribution of each data point with respect to each generated hypothesis, and search the histogram for modes. The residuals will cluster around a structure in the dataset forming the mode. One advantage of the RHA method is that it is an unsupervised approach. The number of inliers per structure, nor the number of structures in the dataset need to be known prior to the search.

The mode search algorithm is presented in three steps [23]:

1. Smooth the histogram with a narrow window and local maxima (modes) and minima (valleys) are located.
2. Remove spurious weak modes and valleys so that only single local minimum valley is present between two modes and only one local maximum mode is present between two valleys.
3. Choose the weakest unlabeled mode and measure its distinctness. If the mode is distance, then it is labeled and added to the list of modes; otherwise it is marked as spurious and removed. If there are no more unlabeled modes, stop the procedure, otherwise go to step 2.

The major drawback to this algorithm is that finding the modes in multimodal data can be difficult and burdensome. In [15], it was pointed out that the peak corresponding to a structure becomes less localized as the point-model distance increases, drowning the rightmost modes in the noise. It was pointed out in [14] that severe outliers and incorrect bandwidth estimates for density estimation can produce false modes and valleys further drowning actual modes in the noise.

Toldo *et. al.* define a “conceptual representation” where each data point is represented by a preferred set in [15]. Essentially an NxM matrix is formed where N represents the number

of points in the set, and M represents the number of minimum sample sets (MSS) formed. If a point belongs to a CS it is assigned a 1, else it is assigned a 0. In this way each column of the matrix represents the characteristic function of the CS of that structure, while each row represents the preferred set of a point. The preferred set is the set of all structures that the point has given consensus. Points that belong to the same structure will cluster in this conceptual space $\{0, 1\}^M$.

The Jaccard distance is defined as

$$d_J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}. \tag{1.4}$$

This distance measures the overlap of set A and B, with a range of 0 for identical sets, to 1 for disjoint sets. Each preference set is given its own cluster, then "the preference set of a cluster is computed as the intersection of the preference sets of its points [15]." The two clusters with the smallest Jaccard distance are replaces with the union of the two clusters. This process is repeated while the smallest Jaccard distance is less than one. The final structure for each cluster is given by a least squares fit.

J-Linkage relies on *a priori* knowledge to the size of structures or a knowledge of and inlier/outlier ratio. It also relies on eliminating outliers due to the fact that they will show up as small clusters and therefore a rejection threshold must be set. By relying on outliers being small clusters, several instances of a single structure must be found to raise it above this threshold.

Chin *et. al.* [14] introduced a method for detecting multiple structures in a dataset that can contain over 90% total outliers. In their method, a novel mercer kernel is defined allowing

statistical machine learning techniques to be applied. Their method also does not require manual input of an inlier noise threshold. The paper refers to their method as Kernel Fitting (KF) and from here on this paper will use the term KF as well.

Proceeding in the same fashion as [23], M structure hypotheses are generated, and the set of residuals $r_i = \{r_1^i, r_2^i, \dots, r_M^i\}$ of each point x_i to the M structures is analyzed. Each set is sorted in ascending order to define,

$$\widetilde{\boldsymbol{\theta}}_i \coloneqq \{\lambda_1^i, \lambda_2^i, \dots, \lambda_M^i\}, \tag{1.5}$$

where λ_j^i is the sorted index value. From this the Ordered Residual Kernel (ORK) [14] between two data points is defined as,

$$k_{\tilde{r}}(x_{i1}, x_{i2}) \coloneqq \frac{1}{Z}\sum_{t=1}^{M/h} z_t \cdot k_{\cap}^t(\widetilde{\boldsymbol{\theta}}_{i1}, \widetilde{\boldsymbol{\theta}}_{i2}), \tag{1.6}$$

where $z_t = \frac{1}{t}$ are the harmonic series and $Z = \sum_{t=1}^{M/h} z_t$ is the $(M/t)^{\text{th}}$ harmonic number. The authors then define the Difference of Intersection Kernel (DOIK),

$$k_{\cap}^t \coloneqq \frac{1}{h}\left(\left|\widetilde{\boldsymbol{\theta}}_{i_1}^{1:\alpha_t} \cap \widetilde{\boldsymbol{\theta}}_{i_2}^{1:\alpha_t}\right| - \left|\widetilde{\boldsymbol{\theta}}_{i_1}^{1:\alpha_t-1} \cap \widetilde{\boldsymbol{\theta}}_{i_2}^{1:\alpha_t-1}\right|\right), \tag{1.7}$$

Where $\alpha_t = th$ and $\alpha_{t-1} = (t-1)h$. The parameter h is a step size that allows the rate of change of intersection from a fictitious inlier threshold. This parameter does not depend on the noise scale but on M. Since the ORK is a Mercer kernel, the input space X is mapped by φ to a inner product space S,

$$\varphi: x \in X \mapsto \varphi(x) = k(x,\cdot) \in S. \tag{1.8}$$

The fact that the kernels satisfy the Mercer condition means the data can be analyzed in the inner product space without explicit transformation.

With the Mercer kernel defined, the authors attempt to remove gross outliers by exploiting the fact that vectors in $\mathbf{B}$ will have high norms if they correspond to inliers and low norms if they are outliers. $\mathbf{B} = [b_1, \dots b_N]$ is the reduced dimension version of $\mathbf{A} = [\varphi(x_1), \dots \varphi(x_N)]$. The distribution of the $\mathbf{B}$ vector norms is bimodal if gross outliers exist, and only contains a single mode if no gross outliers exist. Therefore by defining an inlier/outlier threshold, the gross outliers can be removed from the dataset. The authors give two possible methods for defining a threshold, a 1D Gaussian Mixture Model (GMM) as,

$$f(b) = \sum_{c=1,2} \pi_c \mathrm{N}(b|\mu_c, \sigma_c), \tag{1.9}$$

where N is a Gaussian with mean $\mu_c$and standard deviation σ_c, while π_c is the mixing coefficient. The threshold is then either the point of equal Mahalanobis distance or the average between two means [14]. Another approach is to use,

$$\psi = \rho \max_{i=1,\dots,N} \|b_i\|^2, \tag{1.10}$$

where $\rho = 0.3$ was determined empirically. Every value below the threshold will be considered a gross outlier and removed from the dataset.

To obtain the structures from the gross outlier free dataset, the centered kernel matrix

$$\mathbf{K}' = \tilde{\mathbf{C}}^{\mathrm{T}} \tilde{\mathbf{C}} \tag{1.11}$$

is used, where $\mathbf{C} = [\varphi(y_1), \dots, \varphi(y_{N'})]$ is the un-centered kernel matrix, and $\{y_i\}$ is the set of outlier free points where $N' < N$. $\tilde{\mathbf{C}}$ is obtained by adjusting $\mathbf{C}$ with the empirical mean $\mathbf{C}$. Using Kernel PCA [12], if $\widetilde{\mathbf{K}}' = \mathbf{R}\boldsymbol{\Omega}\mathbf{R}^T$ is the eigenvalue decomposition of $\widetilde{\mathbf{K}}'$, then the first m principal components is given by,

$$\mathbf{P}^m = \tilde{\mathbf{C}}\mathbf{R}^m(\boldsymbol{\Omega}^m)^{-\frac{1}{2}} \tag{1.12}$$

where $\mathbf{R}^m = \mathbf{R}_{:,1:m}$ and $\boldsymbol{\Omega}^m = \boldsymbol{\Omega}_{1:m,1:m}$, using MATLAB® notation. $\mathbf{D}$ can then be clustered in the inner product space where,

$$\mathbf{D} = [d_1, \dots, d_{N'}] = (\boldsymbol{\Omega}^m)^{-\frac{1}{2}}(\mathbf{R}^m)^{\mathrm{T}}. \tag{1.13}$$

The authors chose to use the Normalized Cut (Ncut) algorithm to cluster the data. The data is purposefully over-segmented due to the difficulty of finding a correct thresholding scheme. Finally, a novel structure merging technique is developed to merge the redundant structures and extract the final structure fitting.

Even though KF does not rely on a manual tuning of the noise scale that RANSAC and similar algorithms rely on, the step h of the ORK is not mathematically defined and is set based on the problem set. Also the GMM depends upon a mean, standard deviation, and mixing parameter that must be manually tuned. Therefore KF simply trades which parameters are tuned. A disadvantage of KF, is that it must do a sampling consensus fit twice, due to the fact that it purposefully over-segments during the clustering process. The structure merging algorithm uses LMedS to get an initial fit of each structure, then merges structures based on another defined inlier threshold.

1.3 A Novel Approach

The algorithm proposed in this paper is a flexible framework that can be adapted to varying problem sets with minimal adjustment. It is powerful in that it is an unsupervised process that can discover the number of structures in the dataset without *a priori* knowledge to the number of models or inlier/outlier ratio. The algorithm is also able to handle a minimum sample set of very large dimensionality with minimal computational overhead. This is demonstrated in section 4. It is robust to gross and pseudo outliers. The algorithm also incorporates a novel method for compressing the sampling strategy is described in section 2 that is a significant improvement over typical sampling methods found in the literature that often require the sample number to be many times the size of the dataset.

The approach in this paper relies on two input parameters to find a best fit model for each structure in the dataset; the system noise threshold t and the minimum detectable structure ϕ. The system noise threshold drives the determination of whether a point is an inlier to outlier, while the minimum detectable model drives an initial estimation for the number of structures in the dataset.

Approaching the problem from this perspective produces a trade-off. On one side there is a hard minimum where structures less than this size are considered noise and discarded, however in a real system, parameters can be measured and tuned to find an optimal size. On the other side, the problem is cast as a multivariate problem in which a sampling scheme can be created to ensure all structures in a dataset can be found to a desired probability. This is an advantage and necessary over the typical sampling approach of estimating the inlier to outlier

ratio. In this scenario, the problem is bivariate, so a desired number of outlier free candidate sets Φ can be specified, but whether all structures are represented by these sets cannot be guaranteed.

In reality this minimum structure threshold must be defined anyway due to the nature of the problem domain. In RANSAC the only the best fit is desired, however in a multi-structure case, n best fits are desired. If the MSS of the structure is k, then there are exactly $\binom{n}{k}$ structures in the scene. Unless the correct number of structures is known prior to fitting, it is impossible to recover less without a threshold to declare noise vs. structure. Any unsupervised process *must* have a minimum size threshold whether explicitly stated or not.

With these two system parameters given and a sampling scheme determined, a method is needed for distinguishing unique structures during the sampling process. If $\Phi \geq \phi$ then it can be considered a structure candidate, as it meets the requirement of minimum size, yet it may not be a best fit. A desirable solution is one that is fast, flexible, simple to implement, and robust. By exploiting problem domain information, a distinctive description of a structure can be formed. For instance, a line can be uniquely described by its slope and intercept. These parameters can be used to form a "descriptor" vector $\mathbf{D}$, that is unique to that line. As will be seen in section 4, even complex structures can often be represented by a very simple descriptor vector. Since this vector is problem domain specific, its length and parameters will vary according to problem under investigation. The dimensionality can be reduced by forming a unique label, $\lambda = \|\mathbf{D}\|$. Similar model candidates will tend to cluster around this label and if $\mathbf{D}$ is well formed, the clusters will be very distinct and easily segmented. All valid structures from

the dataset can then be realized by using a clustering scheme. In this thesis, *k*-means [10], [4] was used to find all the clusters. A least-squares fit of each cluster is performed to give the consensus set of the cluster. Due to the fact that a unique label is assigned to each structure candidate, the technique has been named Labeled Sampling Consensus or L-SAC. This name will be used throughout the rest of this thesis.

1.4 Organization Of Thesis

The rest of this paper has been organized into four different sections. Section 2 develops the L-SAC method demonstrating it on the trivial case of finding lines in a dataset containing outliers. A novel sampling scheme is then introduced to dramatically reduce the required samples needed to recover all structures. Section 3 extends L-SAC to finding planes and homographies in a stereo-vision application. Section 4 applies L-SAC to detecting human faces from 3D point cloud data obtained from an Xbox Kinect. Finally Section 5 presents the opportunity for future work and development of L-SAC as well as final conclusions.

2 LABELED SAMPLING CONSENSUS

2.1 Introduction to L-SAC

Labeled Sampling Consensus or L-SAC is based on human decision making and reasoning. For example, suppose a person was trying to survey the opinion of a certain type of person in a crowd of people. First the surveyor would scan the crowd looking for this type of person. When the surveyor finds someone of interest, he or she will form a description of that person such as, “the older gentlemen wearing glasses to my left.” Once the surveyor makes her way to the person of interest, she will ask for a name or “label” to condense this description formed of the person to make it easier to identify him. The surveyor will continue in this fashion until all people of interest have been interviewed for their opinions. During this process, the surveyor may also have to group like individuals as someone may introduce themselves as “Rob”, but may also be referred to as “Robert.”

In this very same fashion, L-SAC will sample the dataset and attempt to find structures of interest. A description of structure candidates found is created to distinguish between different structures of the same class. This description, in the form of a vector, is further reduced to a 1D label to reduce the complexity of the search and identification problem. A well-formed “descriptor” vector will create very distinct boundaries, causing similar labels to cluster close together, and divergent labels to be well separated. This process, allows the structures of

interest to be extracted by counting the number of clusters, and a best fit for the structure can be obtained from the data points within the cluster.

The L-SAC algorithm is now demonstrated using the case of finding multiple lines in a dataset with outliers. A synthetic dataset was generated containing 6 lines, each containing 50 points, and with a range [0 1]. The lines were perturbed by Gaussian noise with 0 mean and a standard deviation of $\sigma = 0.006$. The dataset was then contaminated with 200 points of random outliers. The synthetic data is shown in Figure 2.1.

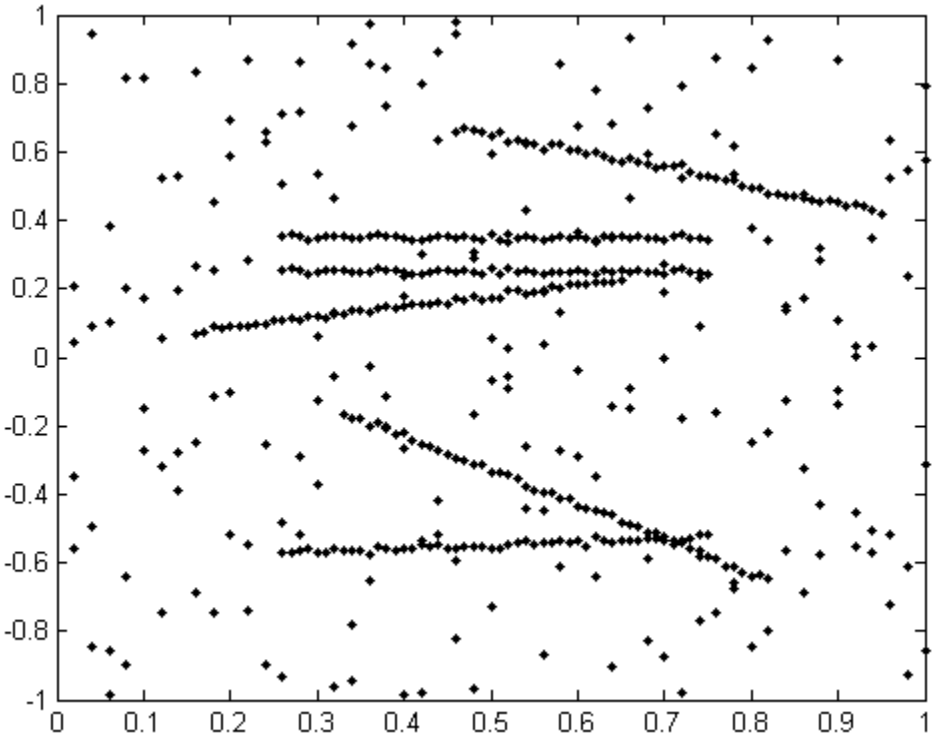

Figure 2.1: Demo dataset of 500 points and 6 lines that meet the minimum structure size ϕ. There are 200 gross outliers, but 90% of the data is an outlier to a single line.

L-SAC depends on four input parameters, the system noise threshold t, minimum model size ϕ, an initial estimate for the number of structures c, and the minimum sample set MSS. For this case, the noise threshold and minimum model size have been determined heuristically to be, $t = 2\sigma$, and $\phi = 50$ respectively. To initially estimate the number of structures, L-SAC will initially assume a worst case scenario, whereby all structures in the system are size ϕ, and all or almost all points are inliers. This scenario is worst case because as the number of structures in the dataset increase, the number of random samples needed increases. This assumption will ensure enough points are sampled, so that all structures in the system are represented in the sampled sets. If n is the total number of points in the dataset, then the initial estimate c is given by,

$$c = \left\lfloor \frac{n}{\phi} \right\rfloor, \tag{2.1}$$

where $\lfloor \quad \rfloor$ represents the floor function. For this example, $n = 500$, so therefore $c = 10$. Finally, the MSS is determined by the structure. In this case it is a line, which can be represented completely by 2 points, therefore the $\mathrm{MSS} = 2$.

Now that the problem set and input parameters are defined, a suitable "descriptor" vector,

$$\mathbf{D} = [a\ b\ c\ \ldots]^{\mathrm{T}} \tag{2.2}$$

and label λ needs to be formed. The idea behind $\mathbf{D}$ and λ is that a dataset may contain multiple structures that are very similar, however they can be distinguished from each other by identifying a set of parameters that can only be contained by a unique instance. For example,

different lines may have the same slope, but they cannot have the same intercept too and still be two unique instances. It is often possible to only require a few parameters to fully distinguish between complex *n*-dimensional structures. This is seen in section 4.2 where human faces are uniquely identified with only a 2 dimensional "descriptor" vector. Every time a candidate set is found, it's unique $\mathbf{D}$ is found by using the model's parameters to create its own vector.

The dimensionality can be reduced by transforming this "descriptor" vector into a 1D label λ. If the parameter selection for $\mathbf{D}$ is good,

$$\lambda = \|\mathbf{D}\| \tag{2.3}$$

will identify all unique instances of similar *n*-dimensional structures in the dataset. The power of the label is that finding any *n*-dimensional structure can be reduced to a 1D clustering problem.

Two dimensional lines present the unique case where the vector $\mathbf{D}$ is a scalar due to the small dimensionality of the structure. The descriptor, $\mathrm{D} = \theta_x + 2b$ was used in this case, where θ_x is the angle of the line to the x-axis. The y-intercept is multiplied by 2 and added to the slope to make the descriptor numerically unique. For example, a line represented by $m = 0$ and $b = 0.5$ is numerically the same as the line $m = 0.5$ and $b = 0$. If the line's slope approaches ∞ the x-intercept is used instead. From D, a label λ can be created to attach to each model candidate set Φ_i. For lines, the label is simply $\lambda = \mathrm{D}$, because the "descriptor" vector is already a scalar.

With a descriptor vector and label, the sampling procedure described later in section 2.2 is used, to find structure candidate sets where $\Phi_i \geq \phi$. These sets are labeled and catalogued in the set $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$, where m is the total number of sets found. This set is not sorted so the index simply represents the order with which a candidate was found. To determine the number of structures in the dataset, it can be clustered using a simple *k*-means clustering algorithm where the number of final cluster centers represent the number of structures. Since the number of clusters is unknown, c is used to initialize the number of clusters. Initial cluster centers are determined by finding an initial center and defining a step size to get the remaining values. The initial center is found by,

$$\mu_1 = \min(\Lambda), \tag{2.4}$$

and the step size is given by,

$$\delta \stackrel{\text{def}}{=} \frac{\lambda_{max} - \lambda_{min}}{c - 1}. \tag{2.5}$$

The set of initial centers is then $\{\mu_1, \mu_2, \dots, \mu_\alpha\}$, where $\mu_i = \mu_{i-1} + \delta$. Every λ_i is compared to each center and is associated to the center which minimizes,

$$r_i = \left\|\lambda_i - \mu_j\right\|^2. \tag{2.6}$$

This process continues until no updates to the cluster centers are made. The final number of centers gives the number of structures in the dataset, and each cluster contains similar candidate sets. Figure 2.2 shows the clusters formed from the L-SAC process performed on the synthetic dataset. The final best fit structures can be extracted by using a least-squares fit of the points in each cluster. This final fit is shown in Figure 2.3 .

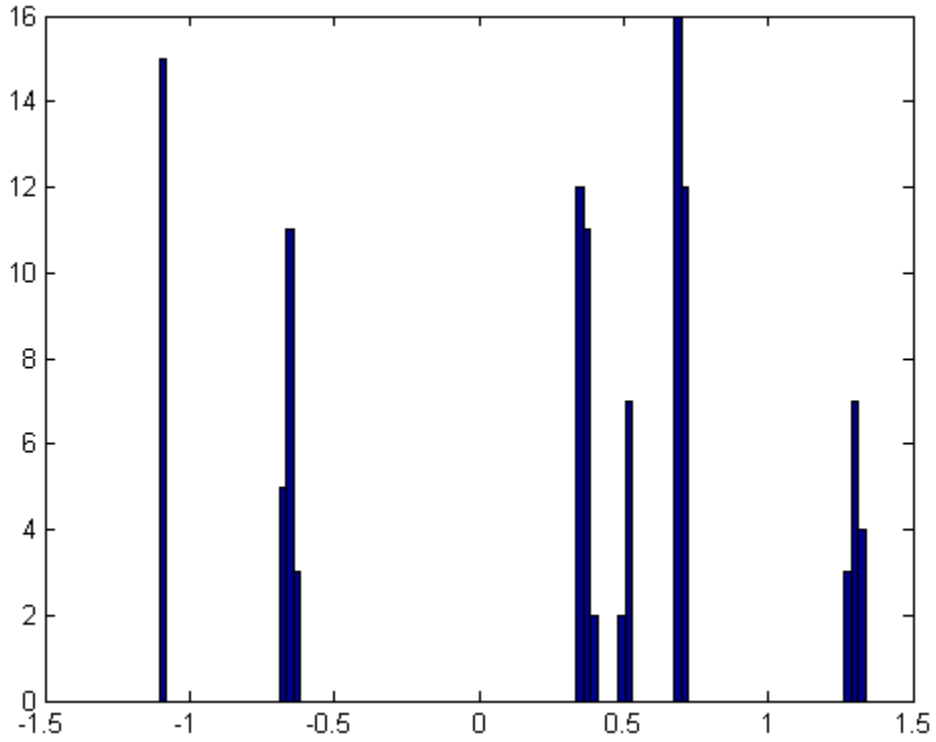

Figure 2.2: It can easily be seen that 6 distinct clusters exist. A good descriptor vector and label will spread the "spectrum" of structure candidates such that distinct boundaries exist.

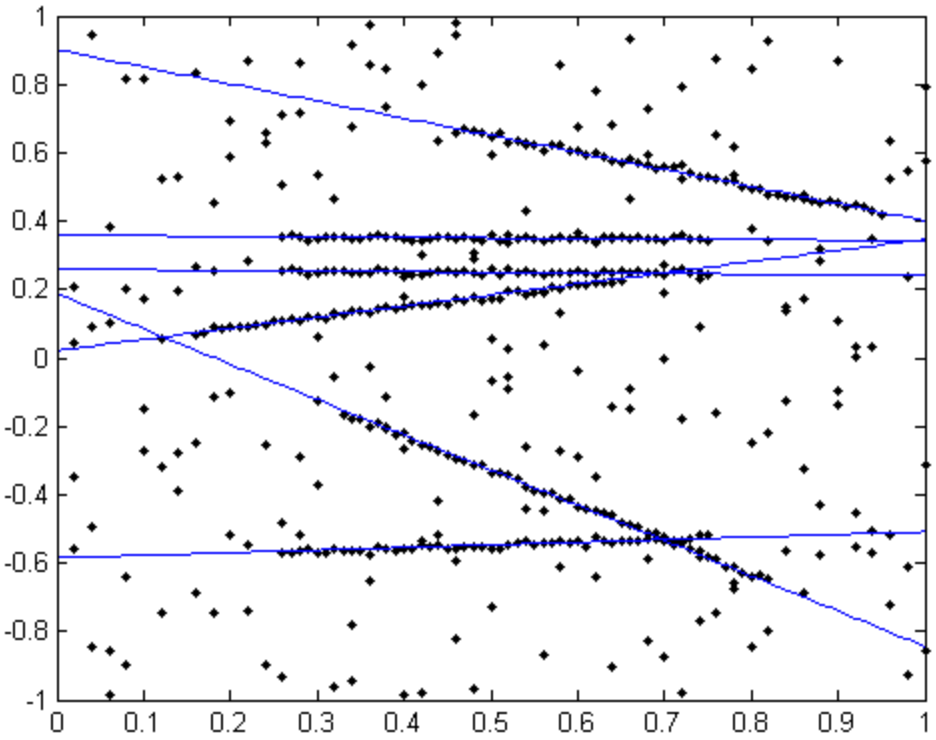

Figure 2.3: The final least-squares fit of all 6 lines which meet the minimum structure size ϕ. This fit was performed with a sample size of only 55 points or 11% of the dataset.

As can be seen L-SAC can be a very effective and robust method for discovering multiple structures in a dataset containing significant outliers, and a relatively large number of structures. Figure 2.3 also shows that L-SAC can distinguish between structures that are very similar as two of the lines have the same slope and were corrupted with the exact same noise. The above example extracted all 6 lines in only one sample of 55 points. Typical papers in the literature required 5000 samples to discover 5 lines. The next section discusses the novel approach to sampling that allows this significant reduction.

2.2 A Novel Sampling Method

When working in a random sampling framework, it is important to understand how many samples are needed to assure finding at least n outlier free structures with a given probability. The problem statement in the literature is most often set up as finding outlier free structures in a known percentage of inliers to outliers. If sampling with replacement is used, then each sample is an independent. If the outlier percentage is ε, then in a RANSAC framework, the probability ρ of finding at least one outlier free structures in N selections is given by [23],

$$\rho = 1 - (1 - (1 - \mathcal{E})^n)^N, \tag{2.7}$$

where n is the number of points needed to instantiate a structure. The probability can be set to an arbitrary value such as 0.99, 0.999, or any value sufficient to the problem set. The required number of samples can then be calculated as,

$$N = \frac{\ln(1-\rho)}{\ln(1-(1-\mathcal{E})^n)}. \tag{2.8}$$

If multiple structures exist in the dataset, then K outlier free structures are desired. If m is the number of Bernoulli trials, the probability of finding K outlier free structures is,

$$\rho = 1 - \sum_{i=0}^{K-1} \binom{m}{i} (1-\mathcal{E})^{ni} (1-(1-\mathcal{E})^n)^{m-i}. \tag{2.9}$$

It was observed by [8] that the probability of selecting an outlier free MSS could be increased by changing the sampling strategy. The first point of the MSS is chosen with a uniform probability while the remaining points are sampled using,

$$P(\mathbf{x}_j|\mathbf{x}_i) = \begin{cases} \frac{1}{Z}\exp^{-\frac{\|\mathbf{x}_j-\mathbf{x}_i\|^2}{\sigma^2}} & \text{if } \mathbf{x}_j \neq \mathbf{x}_i \\ 0 & \text{if } \mathbf{x}_j = \mathbf{x}_i \end{cases}, \tag{2.10}$$

where Z is a normalization constant. This strategy is applied in [25] and [15] to constrain the sampling region for the dependent MSS points $\mathbf{x}_j$ to a Gaussian cluster around $\mathbf{x}_i$. The probability of choosing a MSS with cardinality n of only inliers is,

$$\rho = P(E_1)P(E_2|E_1) \dots P(E_n|E_1, E_2, \dots, E_{n-i}). \tag{2.11}$$

If S is the number of inliers for a given structure, N is the total number of points, α is the average inlier-inlier distance, and ω is the average inlier-outlier distance, the conditional probability can be approximated as [15],

$$P(E_i|E_1, E_2, \dots E_{i-1}) = \frac{(S-i+1)\exp -\frac{\alpha^2}{\sigma^2}}{(N-S-i+1)\exp -\frac{\omega^2}{\sigma^2} + (S-i+1)\exp -\frac{\alpha^2}{\sigma^2}}, \qquad i = 2 \dots n. \tag{2.12}$$

If $S \gg n$, and $\delta = \frac{S}{N}$, then the probability of choosing an outlier free MSS is then,

$$\rho \simeq \delta \left(\frac{\delta exp -\frac{\alpha^2}{\sigma^2}}{(1-\delta) exp -\frac{\omega^2}{\sigma^2} + \delta exp -\frac{\alpha^2}{\sigma^2}} \right)^{n-i}. \tag{2.13}$$

By choosing σ such that a dependent point of arbitrarily close distance $\tilde{d}$, chosen with probability $\tilde{P}$, the conditional probability $P(\mathbf{x}_j|\mathbf{x}_i)$ can get arbitrarily close to 1. Then the effective dimensionality of the manifold reduces; $\mathbb{R}^n \to \mathbb{R}^1$, i.e. $\rho^n \to \rho$.

The main problem that arises from this sampling procedure is that this procedure provides a specified number of outlier free structures, however it does not guarantee that

these structures represent all structures in the data set. Finding multiple structures is a multinomial problem of dimension S^n. The either or approach of the Bernoulli trial creates an ambiguity by reducing the dimension to S^2. Therefore this approach cannot answer the question, "Have all structures in the data set been found?" The only way around this is to multiply the calculated number of samples by an estimate for the number of structures. This number can grow large very quickly if the dataset is large or the number of structures is large.

To overcome this problem, it is proposed that instead of beginning from a known inlier/outlier ratio, the algorithm will begin with a known minimum model size s. This will serve as a threshold for the system to determine if the structure is valid or just noise. With this threshold in place, a worst case estimate for the number of structures in the data set can be determined, i.e. if the data set consists of 1000 points and the minimum detectable structure size is 50, there could be as many as 20 structures that must be found.

This method is more conducive to an unsupervised process versus beginning from a known inlier/outlier ratio, because the only way to know the inlier/outlier ratio is to know the number of structures in the system. In a RANSAC framework, there is only one structure to be discovered; therefore the estimate for the inlier/outlier ratio can be updated during the sampling process. In a multi-structure environment this is impossible as points that are inliers to one structure are probably outliers to another. A set from a structure candidate does not yield information to the number of inliers vs. outliers because of this and because what unique structures have been found is also unknown. Therefore the only way to know the inlier/outlier ratio is to have *a priori* knowledge of the input data.

A novel approach to sampling is presented here with the goal of ensuring all structures will be instantiated and reducing the number of random samples needed to ensure success to a given probability. Typically the MSS of a structure is drawn, processed then replaced back into the population. This process repeats until the desired number of iterations is reached. Sampling with replacement is easier to handle mathematically due to the independence of events and the literature further simplifies things by posing the problem so that it can be represented by a binomial distribution. However, if there are more than one structure and gross outliers, the problem is really a multinomial one. Also, sampling without replacement limits the total number of samples that can be taken. Then, if there was a way that the sampling process can be done without replacement, and an easy way to mathematically describe how many samples need to be taken, the combinations forming the minimum sample sets in the dataset can be realized in an efficient manner.

The novel approach then is, rather than picking an MSS, replacing, and repeating, a one-time "grab" of n samples will be taken. This is sufficient as long as it can be guaranteed that all structures are represented in the "grab" to a specified probability. The minimum sample sets can then be obtained from combinations of the sampled points. It will be shown that often only a small fraction of the dataset needs to be sampled to ensure all structures. It is also noted that it often requires thousands of random samples for multi-structure sampling consensus problems to work. This novel method effectively reduces the sampling process to one sample "grab."

To calculate the required number of samples to be retrieved in the one-time grab, start with finding at least one point on all estimated structures. This scenario can be cast as selecting colored balls from an urn problem. The total number of data points is N, and the minimum detectable structure size is ϕ. The estimated number of structures in the data set is then,

$$c = \begin{cases} \frac{N}{\phi} & \text{if mod}\left(\frac{N}{\phi}\right) = 0 \\ \frac{N}{\phi} + 1 & \text{if mod}\left(\frac{N}{\phi}\right) > 0 \end{cases}. \tag{2.14}$$

It should be noted that the formula for calculating the number of structures is different here than Equation (2.1) because for this situation, if there is a remainder, it counts as an additional structure. Each structure is represented by a set of balls of a unique color, and there is an urn associated to each color. The probability of a particular draw is then given by the multivariate hypergeometric pmf,

$$P(k_1, k_2, \dots, k_c) = \frac{\prod_{i=1}^{c} \binom{m_i}{k_i}}{\binom{N}{n}}, \tag{2.15}$$

where n is the number of points selected and k_i is the number of points selected in structure m_i. To find the needed n for at least one point on every structure with an arbitrary probability requires determining the permutations in the numerator that produce a "correct" outcome, i.e. an outcome with at least one ball in every urn. To find a correct outcome, all the partitions of n into *c* values must be determined. For example, if $n = 5$, it can be partitioned 7 ways,

$$
\begin{array}{c}
5 \\
4\ 1 \\
3\ 2 \\
3\ 1\ 1 \\
2\ 2\ 1 \\
2\ 1\ 1\ 1 \\
1\ 1\ 1\ 1\ 1.
\end{array}
\tag{2.16}
$$

Each partition represents the number of balls in an urn, so the partition 2 2 1 would mean 2 balls in urn 1, 2 balls in urn 2, and 1 ball in urn 3. If the number of structures is 3, than only partitions 3 1 1, and 2 2 1 would be considered correct. Once the correct partitions are found, the number of permutations of each partition must be computed because 2 1 1 is a different draw than 1 2 2. The number of permutations of each partition is,

$$M = \frac{n!}{k_1!\, k_2! \ldots k_c!}. \tag{2.17}$$

A generating function for determining the number of unconstrained partitions of a number is given by the reciprocal of Euler's function [1],

$$\sum_{\tau=0}^{\infty} \mathrm{P}(\tau)\alpha^{\tau} = \prod_{i=1}^{\infty} \left(\frac{1}{1-\alpha^{i}}\right). \tag{2.18}$$

To find the number of correct partitions j, a partitioning algorithm in [24] was modified to find the correct partitions, and calculate j. Therefore, if there are j correct partitions, then the probability of selecting at least one point on every structure given n samples can then be calculated as,

$$\rho = \sum_{i=1}^{j} M_i \frac{\binom{m_1}{k_1}\binom{m_2}{k_2}\cdots\binom{m_c}{k_c}}{\binom{N}{n}}. \tag{2.19}$$

This method works extremely well when the number of structures $c \leq 10$. However, the number of ways to partition a number grows quickly. For instance, the number 10 can be partitioned 42 ways, while the number 20 can be partitioned 89,134 ways [13]. The computational complexity from solving the probability this way grows rapidly to the point of impossibility.

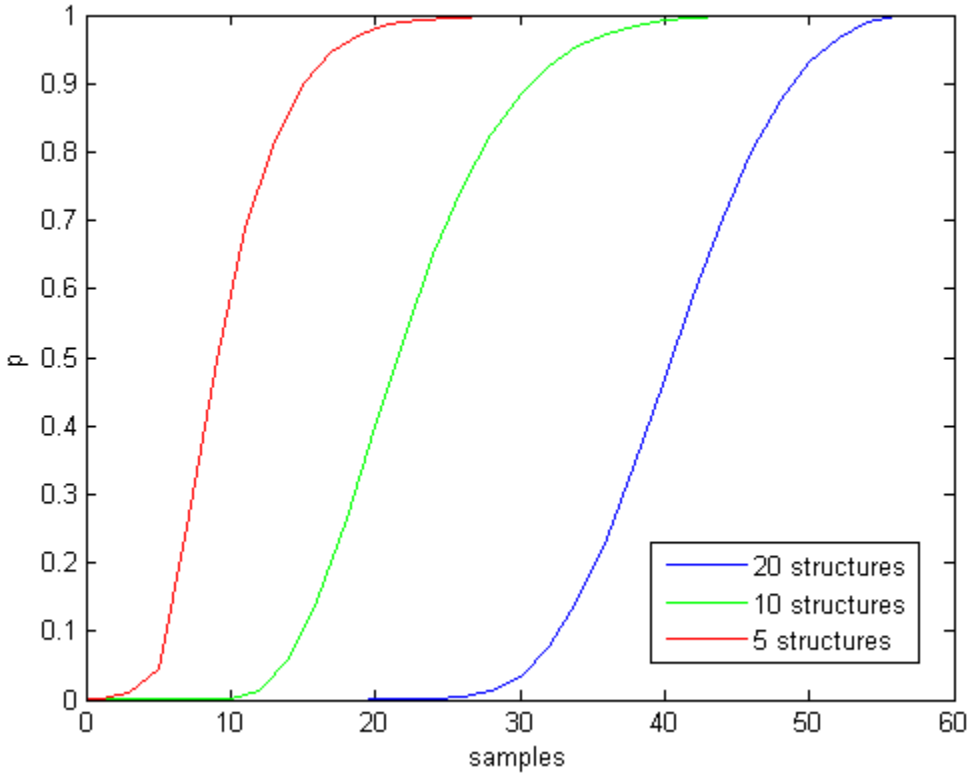

Figure 2.4: This graph shows the cdf of picking at least one point on every structure for the case of 5, 10, and 20 structures in a dataset of 60. All curves were calculated exactly.

To overcome this issue, a statistical approach can be applied to the problem. If there are c structures estimated in the dataset and n points sampled at a time, the question that must be answered is, "has a point on all structures been found?" This yes or no question can be described by a Bernoulli distribution where '1' means yes, and '0' means no. If there are t Bernoulli trials then $\mathbf{x}_t = (x_1, x_2, \dots, x_t)$ represents the outcome of each experiment. If ρ is the probability of success or '1', then the pmf for any outcome is,

$$p(x_k|\rho_s) = \rho_s^{x_k}(1-\rho_s)^{1-x_k} = \begin{cases} \rho_s & \text{if } x_k = 1 \\ 1-\rho_s & \text{if } x_k = 0 \end{cases}. \tag{2.20}$$

The log likelihood function $\ln L(x_1,\ x_2, \dots, x_k; \hat{\rho})$ gives,

$$\sum_{i=1}^{t} (x_i \ln \hat{\rho} + (1-x_i)\ln(1-\hat{\rho}), \tag{2.21}$$

where $\hat{\rho}$ is the parameter that maximizes the likelihood function. Taking the first derivative and setting it equal to zero yields,

$$0 = -\frac{t}{1-\hat{\rho}} + \frac{1}{\hat{\rho}(1-\hat{\rho})}\sum_{i=1}^{t} x_i\,. \tag{2.22}$$

Finally the maximum likelihood estimate is,

$$\hat{\rho} = \frac{1}{t}\sum_{i=1}^{t} x_i. \tag{2.23}$$

It is easily seen that the probability of selecting at least one point on every structure given n samples is simply the relative frequency of t Bernoulli trials. It can also be seen that $\hat{\rho}$ is an unbiased estimator [9]. Arbitrarily specifying a desired number of points on a given structure

is now a trivial task of changing the question to, "have $\hat{n}$ points on every model been found?" This is still a Bernoulli trial and can be calculated exactly the same as above. Figure 2.7 shows a plot of the cdf for at least 5 points and 10 points being picked on every structure to any arbitrary probability.

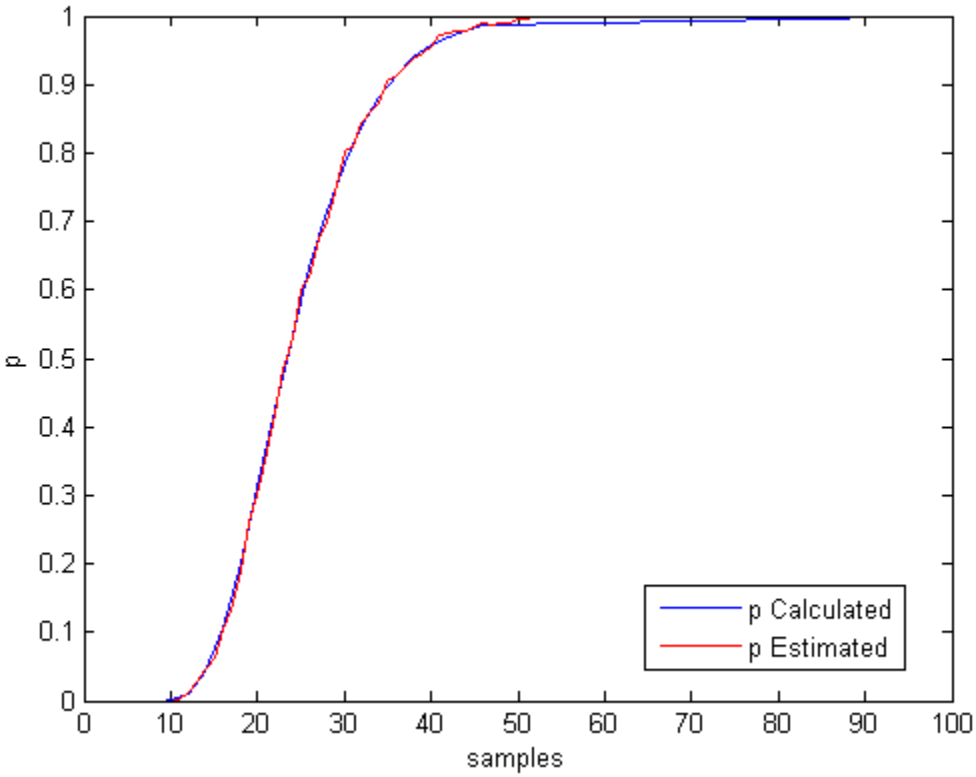

Figure 2.5: The graph shows the cdf of picking at least one point on every structure in the case of 10 structures in a dataset of 100. The blue curve is calculated exactly, while the red curve is estimated. As can be seen the curves are almost identical.

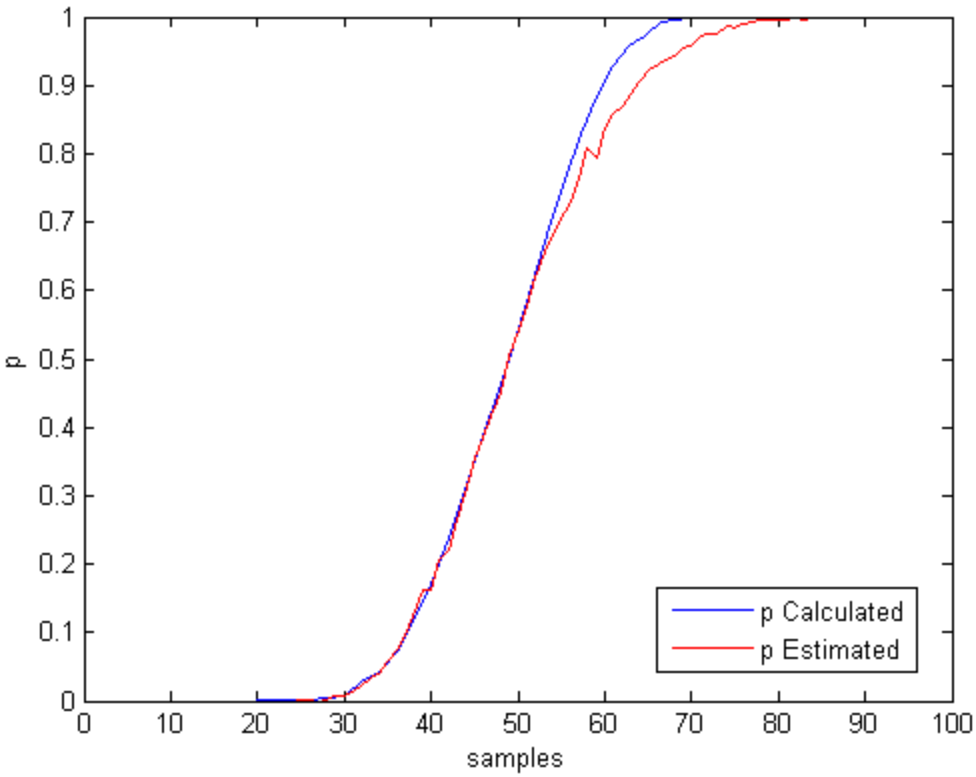

Figure 2.6: The graph shows the cdf of picking at least one point on every structure for the case of 20 structures in a datset of 100 points. The blue curve is calculated exactly. Due to computational overhead, only the curve up to a p of 0.5 was calculated, then reflected about this point. The red curve shows the estimated cdf and very closely matches the blue curve.

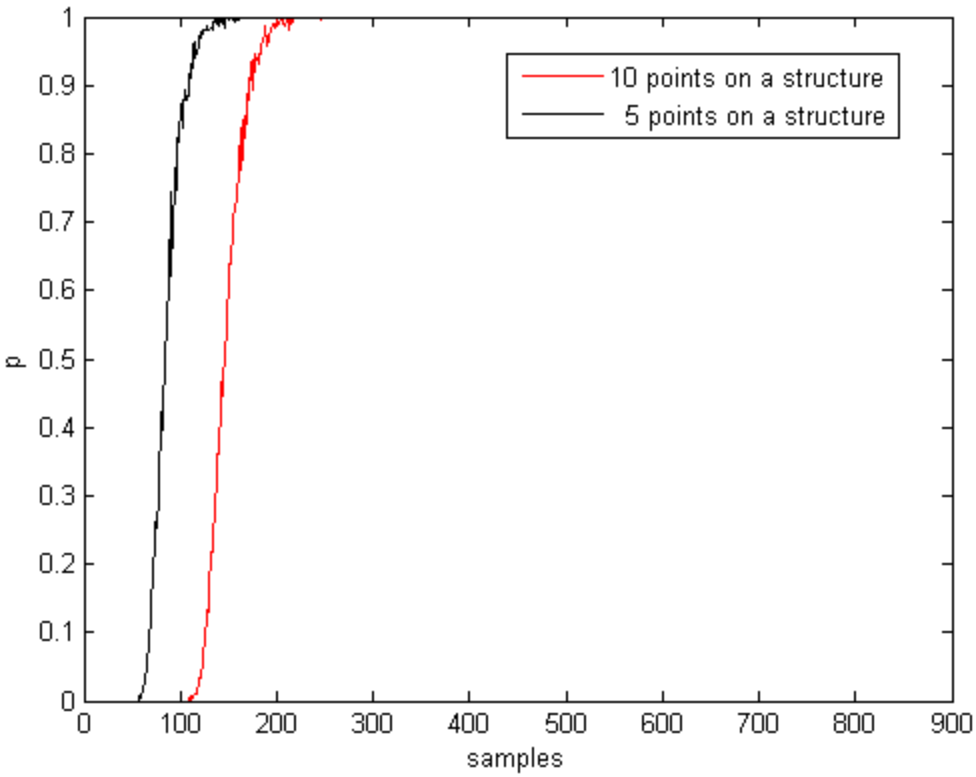

Figure 2.7: A plot of the cdf for the number of samples needed to ensure $\hat{\boldsymbol{n}}$ points on each structure to an arbitrary probability. The hypothetical dataset is 900 points containing up to 10 structures.

2.3 Application of L-SAC to Finding 2D Lines

Another test case, this time involving 10 lines was examined using L-SAC. The size of the dataset was 700 points, and 10 lines of 50 points each were corrupted with Gaussian noise $\sigma = 0.006$. Only 95 points were sampled as shown in Figure 2.8. Figure 2.9 shows the 10 clusters found via L-SAC, while Figure 2.10 shows the final least-squares fit.

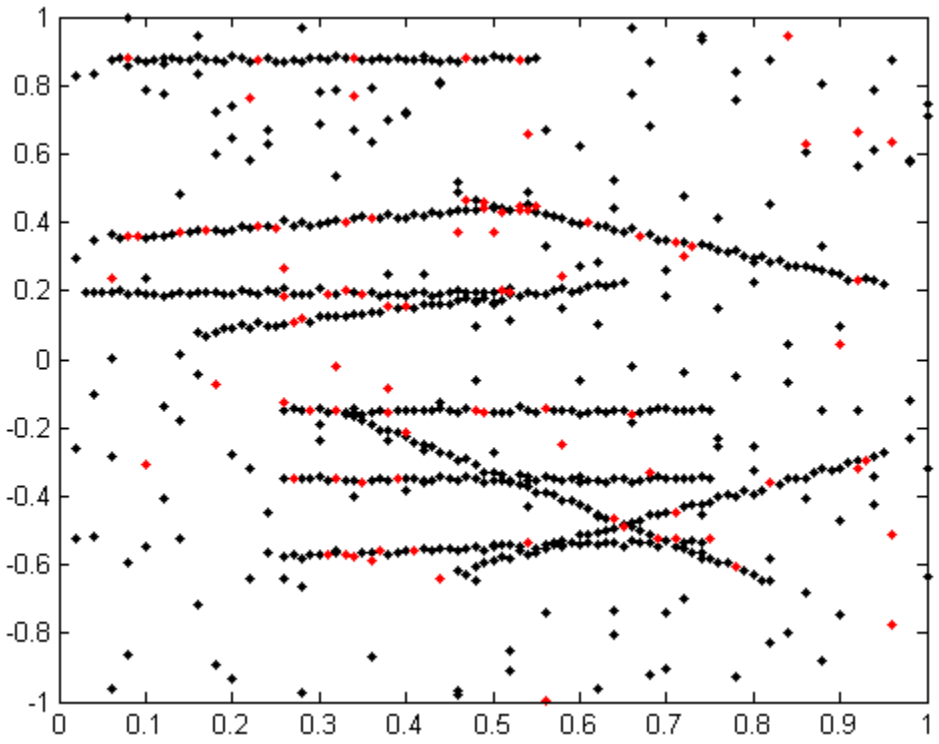

Figure 2.8: Ninety-five points, shown in red were sampled from the dataset of 700.

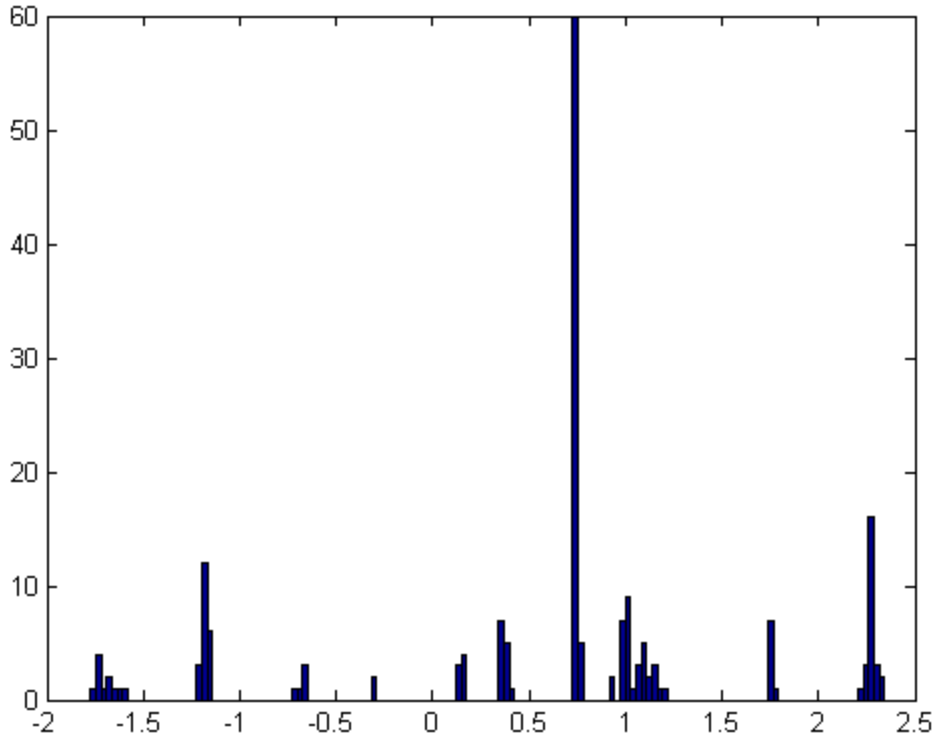

Figure 2.9: The 10 clusters are clearly distinct.

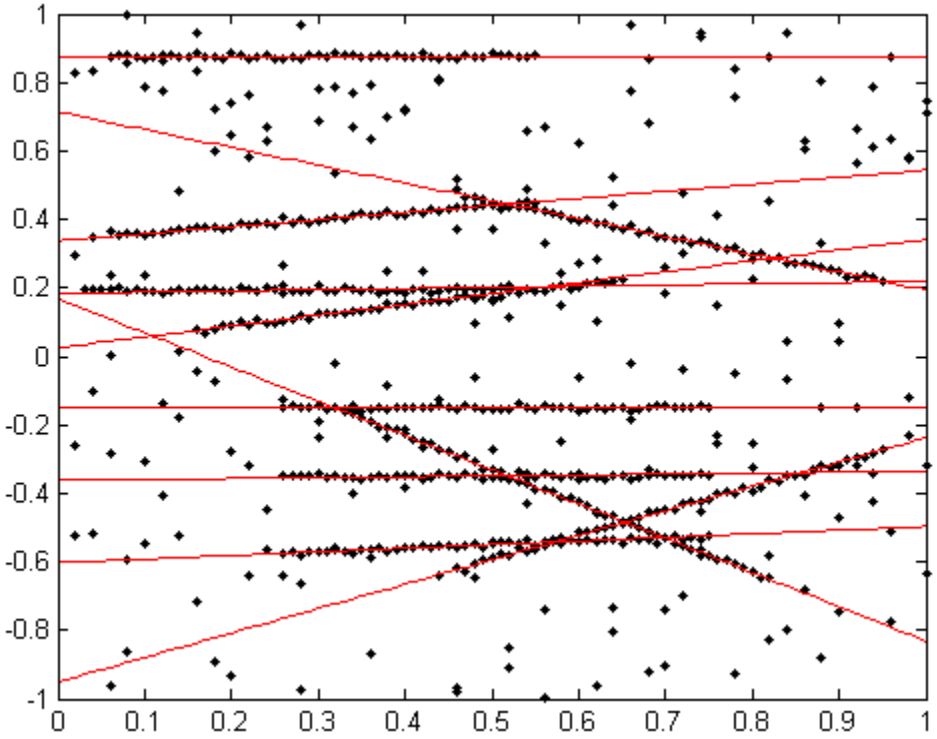

Figure 2.10: All 10 lines fitted using a least-squares fit from each cluster.

3 L-SAC AND FINDING MULTIPLE PLANES

3.1 Application to Planes and Homographies in Stereo Vision

Finding multiple structures using L-SAC can easily be extended to finding homographies and scene planes in two-view geometry. Hartley *et al.* [7] discuss the two methods for finding the homography $\mathbf{H}$ induced by three non-collinear points $\mathbf{x}_i$ and the fundamental matrix $\mathbf{F}$. The first method is to explicitly reconstruct the world point from the imaged point correspondences. Suppose $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ are 3 homogeneous points where $\mathbf{x}_i = [x_i \; y_i \; z_i \; 1]^{\mathrm{T}}$, then the plane $\mathbf{p}$ can be found by,

$$\mathbf{p} = \begin{bmatrix} (\mathbf{x}_1 - \mathbf{x}_3) \times (\mathbf{x}_2 - \mathbf{x}_3) \\ -\mathbf{x}_3{}^{\mathrm{T}}(\mathbf{x}_1 \times \mathbf{x}_2) \end{bmatrix}. \tag{3.1}$$

The second method involves solving for the homography implicitly. Three point correspondences are related by $\mathbf{x}_i' = \mathbf{H}\mathbf{x}_i$, and the fourth can be obtained via $\mathbf{e}' = \mathbf{H}\mathbf{e}$, where $\mathbf{e}'$ and $\mathbf{e}$ are the epipoles from each view. With 4 point correspondences the homography can be computed using the Direct Linear Transform (DLT) algorithm. The explicit approach was chosen as the implicit approach "has significant degeneracies which are not present in the explicit method [7]."

Given 2 input images from 2 different views of the same scene, and point correspondences $\{ \mathbf{x}_i \leftrightarrow \mathbf{x}_i' \}$, the fundamental matrix $\mathbf{F}$ was computed using the normalized 8-point algorithm from [7]. First the points are normalized such that the centroid of the points is

the origin, and the RMS distance is $\sqrt{2}$. This will condition the points to improve the performance. Every point is multiplied by the transformation matrix

$$\mathbf{T} = \begin{bmatrix} \sigma & & -\sigma c_x \\ & \sigma & -\sigma c_y \\ & & 1 \end{bmatrix}, \tag{3.2}$$

where $\mathbf{c} = [c_x \; c_y \; 1]^{\mathrm{T}}$ are the homogeneous coordinates of the centroid, and σ is the scaling factor such that $d(RMS)s = \sqrt{2}$. $\mathbf{F}$ is obtained by taking the Singular Value Decomposition (SVD) of the constraint matrix

$$\widehat{\mathbf{A}} = \begin{bmatrix} x_1'x_1 & x_1'y_1 & x_1' & y_1'x_1 & y_1'y_1 & y_1' & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n'x_n & x_n'y_n & x_n' & y_n'x_n & y_n'y_n & y_n' & x_n & y_n & 1 \end{bmatrix}. \tag{3.3}$$

It is important to force $\mathbf{F}$ to have rank-2, so that $\det \widehat{\mathbf{F}'} = 0$ by taking the SVD of $\mathbf{F}$ and rebuilding $\widehat{\mathbf{F}'}$ with the 2 largest singular values. By de-normalizing, $\mathbf{F}$ can be found as $\mathbf{F} = \mathbf{T}'^{\mathrm{T}}\widehat{\mathbf{F}'}\mathbf{T}$.

With the fundamental matrix found, the camera model pair $\widehat{\mathbf{P}} = [\,\mathbf{I}\,|0\,]$ and $\widehat{\mathbf{P}'} = [\,[\mathbf{e}']_{\mathrm{x}}\mathbf{F} \mid \mathbf{e}'\,]$ can be determined where $[\mathbf{e}']_{\mathrm{x}}$ is a skew-symmetric matrix and $\mathbf{e}'$is an epipole. Cameras $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{P}'}$ are projective cameras, so a scene reconstruction would have projective distortion. A Euclidean reconstruction was desired, so the camera calibration matrix

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \tag{3.4}$$

is needed. The intrinsic parameters α_x and α_y are to focal length, s is the skew, and $\mathbf{x}_0 = [x_0\ y_0]^{\mathrm{T}}$ is the principal point. The values for the matrix $\mathbf{K}$ were determined using Zhang's calibration method [21] and executable provided at [22].

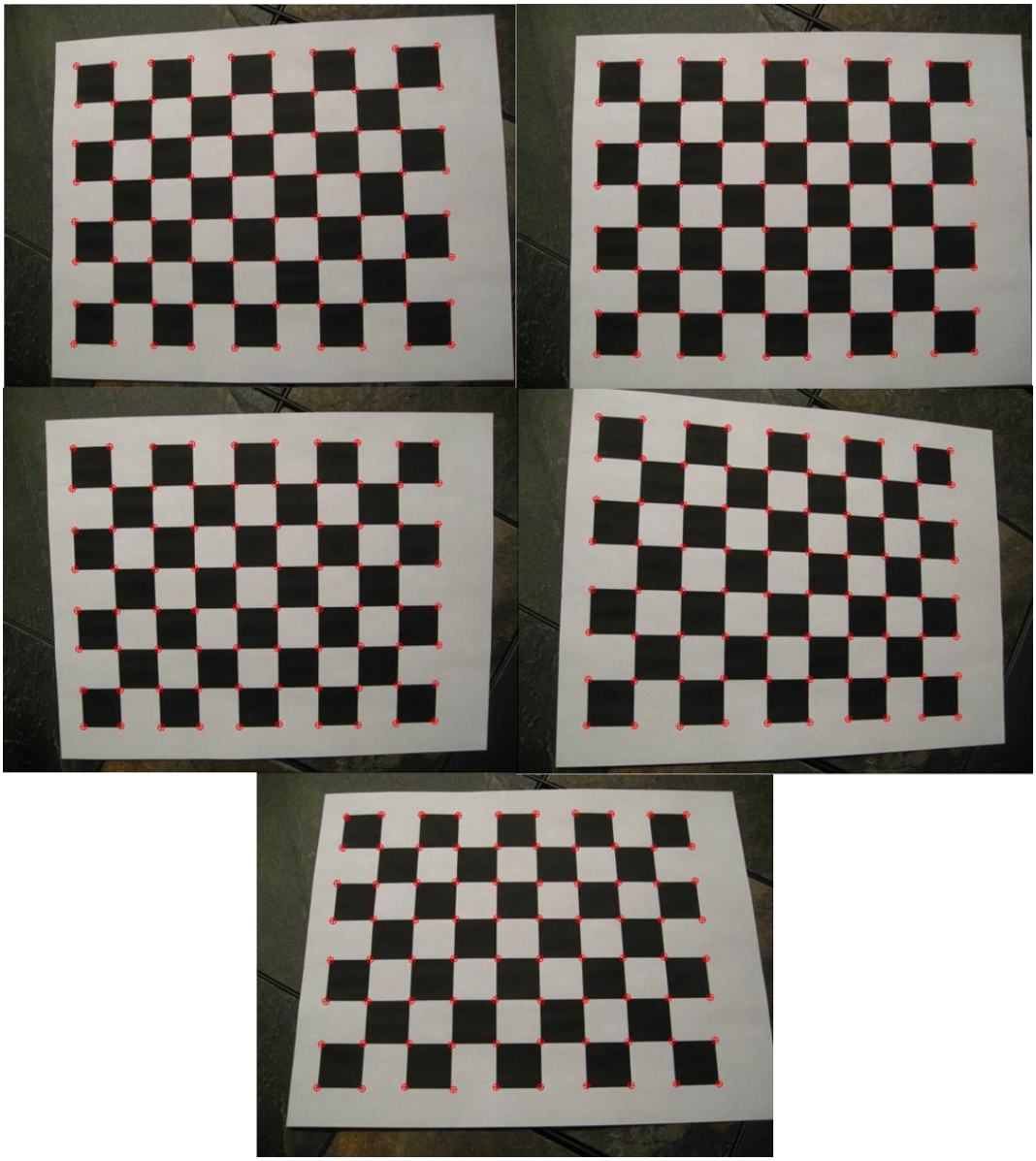

Figure 3.1: Five images from different angles of a single checkerboard were taken to calibrate the camera. The point matches are shown in red.

The Euclidean camera $\mathbf{P} = \mathbf{K}\widehat{\mathbf{P}}$ is easily found however $\mathbf{P}'$ requires the essential matrix. The essential matrix $\mathbf{E}$, is a specialized case of the more general fundamental matrix, in which the image coordinates are normalized. It is given by,

$$\mathbf{E} = [\mathbf{t}]_{\mathrm{x}}\mathbf{R} = \mathbf{K}^{\mathrm{T}}\mathbf{F}\mathbf{K}. \tag{3.5}$$

Taking the SVD of $\mathbf{E}$ yields $\mathbf{U}\mathrm{diag}(1, 1, 0)\mathbf{V}^{\mathrm{T}}$. Now suppose there are matrices,

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{Z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3.6}$$

Then $\mathbf{E}$ may be factored as $\mathbf{E} = \mathbf{SR}$, where $\mathbf{S}$ is a general skew-symmetric matrix, and $\mathbf{S}$ can be written as $\mathbf{S} = k\mathbf{U}\mathbf{Z}\mathbf{U}^{\mathrm{T}}$. The term k means that a camera $\mathbf{P}'$can be extracted from the essential matrix up to a scale. There are then two possible factorizations for $\mathbf{E} = \mathbf{SR}$,

$$\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}} \textit{ or } \mathbf{R} = \mathbf{U}\mathbf{W}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}}. \tag{3.7}$$

Since $\mathbf{S}\mathbf{t} = \mathbf{0}$, $\mathbf{t} = \mathbf{U}[0\ 0\ 1]^{\mathrm{T}} = \mathbf{u}_3$, where $\mathbf{u}_3$ is the last column of $\mathbf{U}$. This means there are four possible solutions for camera $\mathbf{P}'$ given the essential matrix $\mathbf{E}$. They are,

$$\begin{aligned} \mathbf{P}_1' &= [\mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}} \mid +\mathbf{u}_3], \\ \mathbf{P}_2' &= [\mathbf{U}\mathbf{W}\mathbf{V}^{T} \mid -\mathbf{u}_3], \\ \mathbf{P}_3' &= [\mathbf{U}\mathbf{W}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}} \mid +\mathbf{u}_3], \\ \mathbf{P}_4' &= [\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}} \mid -\mathbf{u}_3]. \end{aligned} \tag{3.8}$$

Finally the Euclidean camera is obtained by $\mathbf{P}' = \mathbf{K}\mathbf{P}_i'$, where $\mathbf{P}_i'$ is the solution that puts the scene in front of both cameras.

With the Euclidean canonical camera pair, the imaged point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i'\}$ can be back projected to their 3D world points, where the points are unambiguous up to a scale. To perform the back projection and find the homogeneous world points $\mathbf{w} = [x\ y\ z\ 1]^{\mathrm{T}}$, the optimal triangulation method found in [7] was used. First a point pair $\mathbf{x}_i$ and $\mathbf{x}_i'$ are translated to the respective origins by,

$$\mathbf{T} = \begin{bmatrix} 1 & & -x \\ & 1 & -y \\ & & 1 \end{bmatrix} \text{ and } \mathbf{T}' = \begin{bmatrix} 1 & & -x' \\ & 1 & -y' \\ & & 1 \end{bmatrix}. \tag{3.9}$$

Then the fundamental matrix is replaced by the translated matrix $\mathbf{T}'\mathbf{F}\mathbf{T}^{-1}$. Next the right and left epipoles $\mathbf{e} = [e_1\ e_2\ e_3]^{\mathrm{T}}$s and $\mathbf{e}' = [e_1'\ e_2'\ e_3']^{\mathrm{T}}$ are found where $\mathbf{e}'^{\mathrm{T}}\mathbf{F} = \mathbf{0}$ and $\mathbf{F}\mathbf{e} = \mathbf{0}$. These points are normalized so that $e_1^2 + e_2^2 = 1$ and $e_1'^2 + e_2'^2 = 1$. Using the rotation matrices:

$$\mathbf{R} = \begin{bmatrix} e_1 & e_2 & \\ -e_2 & e_1 & \\ & & 1 \end{bmatrix} \text{ and } \mathbf{R} = \begin{bmatrix} e_1' & e_2' & \\ -e_2' & e_1' & \\ & & 1 \end{bmatrix}, \tag{3.10}$$

$\mathbf{F}$ is replaced by $\mathbf{R}'\mathbf{F}\mathbf{R}$. The rotation also puts the epipoles $\mathbf{e}$ and $\mathbf{e}'$ on the *x*-axis. This gives the fundamental matrix the form,

$$\mathbf{F} = \begin{bmatrix} e_3e_3'd & -e_3'c & -e_3'd \\ -e_3b & a & b \\ -e_3d & c & d \end{bmatrix}. \tag{3.11}$$

The polynomial,

$$g(t) = t((at+b)^2 + e_3'^2(ct+d)^2)^2 - (ad-bc)(1+e_3^2t^2)^2(at+b)(ct+d) = 0, \tag{3.12}$$

is formed and its 6 roots are found. The real parts of each root is evaluated at a cost function,

$$s(t) = \frac{t^2}{1 + e_3^2 t^2} + \frac{(ct + d)^2}{(at + b)^2 + e_3'^2 (ct + d)^2}, \tag{3.13}$$

and the t which provides the minimum value to this function is selected as t_r. The value $t = \infty$ is evaluated at the asymptotic value,

$$\frac{1}{e_3^2} + \frac{c^2}{(a^2 + e_3'^2 c^2)}. \tag{3.14}$$

Then $t_{min}$is selected where,

$$t_{min} = \min(t_r, t_\infty). \tag{3.15}$$

The two epipolar lines $\mathbf{l} = [te_3^2 \quad 1 \quad -t]^{\mathrm{T}}$ and $\mathbf{l}' = \mathbf{F}[0 \quad t \quad 1]^{\mathbf{T}}$ are evaluated at the t_{min}. For a general line $[a \quad b \quad c]^{\mathrm{T}}$, the closest point on a line to the origin is $[-ac \quad -bc \quad a^2 + b^2]^{\mathrm{T}}$. Evaluating this for $\mathbf{l}$ and $\mathbf{l}'$ gives this points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$. These points are then transferred back to the original coordinate system replacing $\hat{\mathbf{x}}$ with $\mathbf{T}^{-1}\mathbf{R}^{\mathrm{T}}\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ with $\mathbf{T}'^{-1}\mathbf{R}'^{\mathrm{T}}\hat{\mathbf{x}}'$. The 3D world point can now be computed using a linear triangulation method. Since $\hat{\mathbf{x}} = \mathbf{P}\mathbf{w}$ and $\hat{\mathbf{x}}' = \mathbf{P}'\mathbf{w}$, the equations can be formed into a matrix $\mathbf{M}$ such that $\mathbf{M}\mathbf{w} = \mathbf{0}$. If the rows of $\mathbf{P}$ and $\mathbf{P}'$ are the vectors $\mathbf{p}_i^{\mathrm{T}}$ and $\mathbf{p}_i'^{\mathrm{T}}$ respectively, then $\mathbf{M}$ is,

$$\mathbf{M} = \begin{bmatrix} x\mathbf{p}_3^{\mathrm{T}} - \mathbf{p}_1^{\mathrm{T}} \\ y\mathbf{p}_3^{\mathrm{T}} - \mathbf{p}_2^{\mathrm{T}} \\ x'\mathbf{p}_3'^{\mathrm{T}} - \mathbf{p}_1'^{\mathrm{T}} \\ y'\mathbf{p}_3'^{\mathrm{T}} - \mathbf{p}_2'^{\mathrm{T}} \end{bmatrix}. \tag{3.16}$$

Then if $\mathbf{UDV} = \mathrm{svd}(\mathbf{M})$, The world coordinate $\mathbf{w}$ is found by taking the last column of $\mathbf{V}$**.** The final step is to normalize $\mathbf{w}$ it has the form $\mathbf{w} = [x \; y \; z \; 1]^{\mathrm{T}}$. After projecting all points in the

dataset to their real world counterpart, a new set $\mathrm{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$ representing the real world coordinates is formed.

A plane requires a minimum sample set of three points. By using the sampling procedure in section 2.2, the plane equation for each MSS_i can be found, and the number of inliers to this plane counted. To determine if a point $\mathbf{w}_j$ is an inlier to plane $\mathbf{p}_i = [a_i\ b_i\ c_i\ d_i]^{\mathrm{T}}$, the normal to the plane is easily taken as $\mathbf{n}_i = [a\ b\ c]^{\mathrm{T}}$, then converted to the unit normal,

$$\widehat{\mathbf{n}}_i = \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|}. \tag{3.17}$$

The distance from point $\mathbf{w}_j$ to plane $\mathbf{p}_i$ is given by,

$$d = \widehat{\mathbf{n}}_i \cdot (\mathbf{w}_j - \mathbf{w}_0), \tag{3.18}$$

where $\mathbf{w}_0$is any point on plane $\mathbf{p}_i$. If d is less than a heuristically defined noise threshold t, than $\mathbf{w}_j$ is considered an inlier to MSS_i. The total number of inliers is then, $\tau = \sum \mathbf{w}_j < t$. If τ is greater than the minimum detectable plane ϕ, where ϕ is also heuristically chosen, then $\mathbf{p}_i$ is considered a plane candidate.

To distinguish between different plane candidates, it is important to label each candidate with a description that can clearly distinguish it from different candidates, while at the same time cluster with similar candidates. A suitable descriptor vector can be created using the uniqueness of the plane's normal vectors, or more specifically their angles to the *x, y, z*-coordinate planes. It is important to ensure that the normal vectors are oriented consistently or similar plane candidates could get mapped to different labels. A simple method to ensure proper orientation is to multiply $\mathbf{n}$ by -1 if its *z* component is negative, i.e.

$$\mathbf{n} = \begin{cases} \mathbf{n} \text{ if } \mathbf{n} = [a \quad b \quad c]^{\mathrm{T}} \\ -\mathbf{n} \text{ if } \mathbf{n} = [a \quad b \quad -c]^{\mathrm{T}} \end{cases}. \tag{3.19}$$

The angle between $\mathbf{n}$ and the normal to the *x*-coordinate plane is,

$$\theta_x = \cos^{-1} \frac{\mathbf{n} \cdot \mathbf{x}}{\|\mathbf{n}\|\|\mathbf{x}\|}, \tag{3.20}$$

where $\mathbf{x} = [1\ 0\ 0]^{\mathrm{T}}$. If θ_y and θ_z are found in similar fashion, where $\mathbf{y} = [0\ 1\ 0]^{\mathrm{T}}$ and $\mathbf{z} = [0\ 0\ 1]^{\mathrm{T}}$, then a complete description of the plane orientation is given. To further distinguish between similarly oriented planes, the constant d from the general plane equation can be used. The descriptor vector for plane candidate $\mathbf{p}_i$ can therefore be described as,

$$\mathbf{D_i} = [\theta_{ix}\ \theta_{iy}\ \theta_{iz}\ d_i]^{\mathrm{T}}. \tag{3.21}$$

To discover the number of planes in the set of plane candidates found from random sampling, the vector $\mathbf{D}$ is used. The dimensionality of the problem can be reduced from $\mathbb{R}^4$ to $\mathbb{R}^1$, by evaluating,

$$\lambda_i = \|\mathbf{D}_i\|^2. \tag{3.22}$$

Therefore if k plane candidates are found during random sampling, the set $\lambda = \{\lambda_1, \lambda_2, \ldots\ \lambda_k\}$ is formed. The similar candidates will tend to cluster together, and if $\mathbf{D}$ is a good descriptor, the boundaries of the clusters will be distinct. By using *k*-means, the number of planes in the dataset can be determined. If the cardinality of the set W is denoted |W|, then the initial estimate for the number of means is the maximum potential planes in the dataset given by,

$$c = \left\lfloor \frac{|\mathrm{W}|}{\phi} \right\rfloor. \tag{3.23}$$

To ensure that the initial mean values, $\mu_1 \ldots\ \mu_m$ are spaced properly, a step size is defined as,

$$\delta \stackrel{\text{def}}{=} \frac{\lambda_{max} - \lambda_{min}}{c - 1}, \tag{3.24}$$

The *k*-means algorithm will iterate associating each candidate to a mean value. The number of means and mean values will continue to be adjusted until no change is made in association. The number of distinct planes in the dataset is then given by the number of clusters formed. From here, a best fit of the planes discovered in the dataset can be found. A least-squares fit is performed on the points in each cluster j to find $\mathbf{p}_j$.

To recover the homography induced by the recovered planes, 3 imaged point correspondences from the plane, the Fundamental Matrix, and epipoles can be used. The homography is given by

$$\mathbf{H} = \mathbf{A} - \mathbf{e}'(\mathbf{M}^{-1}\mathbf{b})^{\mathrm{T}}, \tag{3.25}$$

where $\mathbf{A} = [\mathbf{e}']_{\mathrm{x}}\mathbf{F}$ and $\mathbf{b}$ is a 3-vector

$$b_i = \frac{\left(\mathbf{x}_i' \times (\mathbf{A}\mathbf{x}_i)\right)^{T}(\mathbf{x}_i' \times \mathbf{e}')}{\left\|\mathbf{x}_i' \times \boldsymbol{e}'\right\|^2}, \tag{3.26}$$

and $\mathbf{M}$ is a 3 x 3 matrix with rows $\mathbf{x}_i^{\mathrm{T}}$ [7].

3.2 Multiple Planes Example

An example was set up to test the performance of L-SAC in finding planes and homographies. A Canon Powershot SD850-IS was set to a resolution of 640x480 and calibrated using Zhang's method. Two-views of the same scene were then captured and 135 point matches representing 3 planes were extracted as shown in Figure 3.2. Only 150 combinations of

24 points were used, yet all planes were recovered. The homography was recovered using Equation (3.25). The total runtime in MATLAB® was 3.16 seconds.

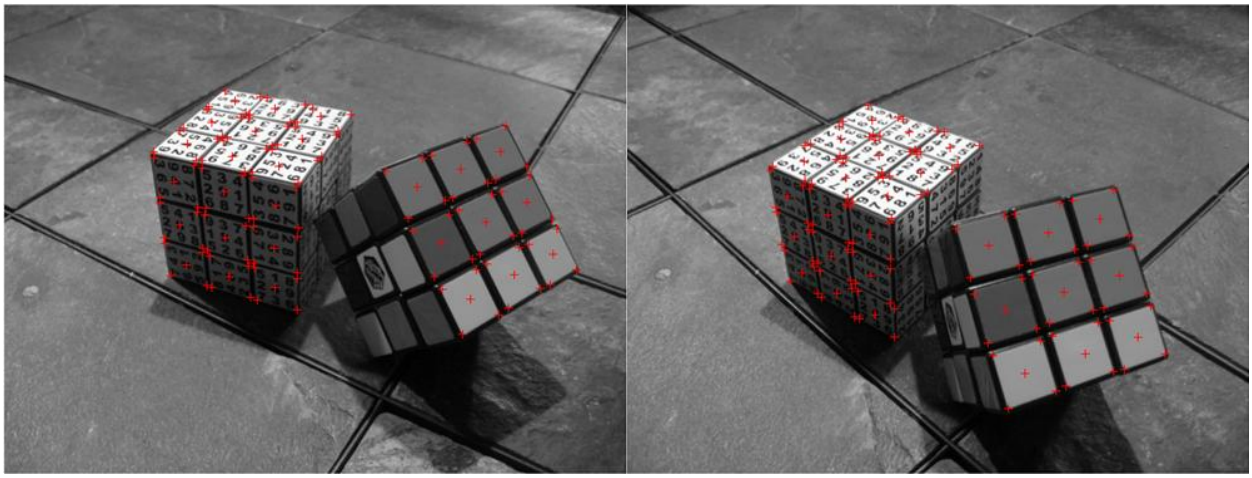

Figure 3.2: Two-views of the same scene were taken. The camera was calibrated using Zhang's method, and the parameters were found to be: $\alpha_x = 672.211$, $\alpha_y = 674.262$, $s = 1.21503$, $x_0 = 326.691$, and $y_0 = 242.046$. The point matches are shown in red and compose 3 image planes.

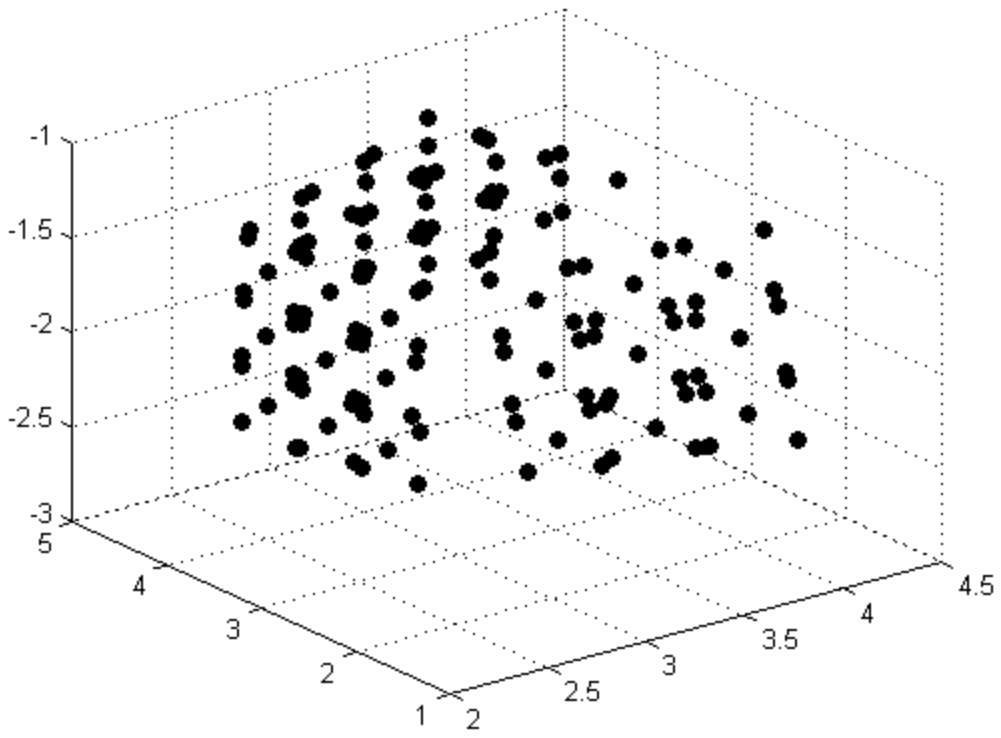

Figure 3.3: Point matches back-projected into 3D world points.

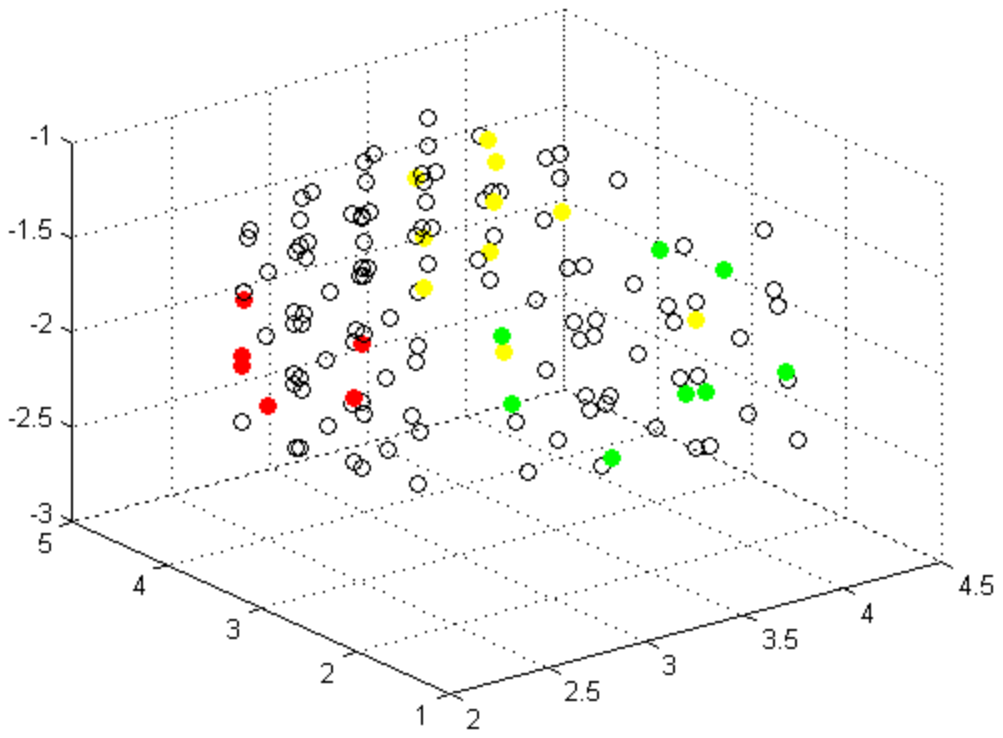

Figure 3.4: The points used to find the plane are overlayed the back-projected points. The 3 colors represent the 3 different clusters found. As can be seen, only 24 points were used, and only 150 combinations were tried.

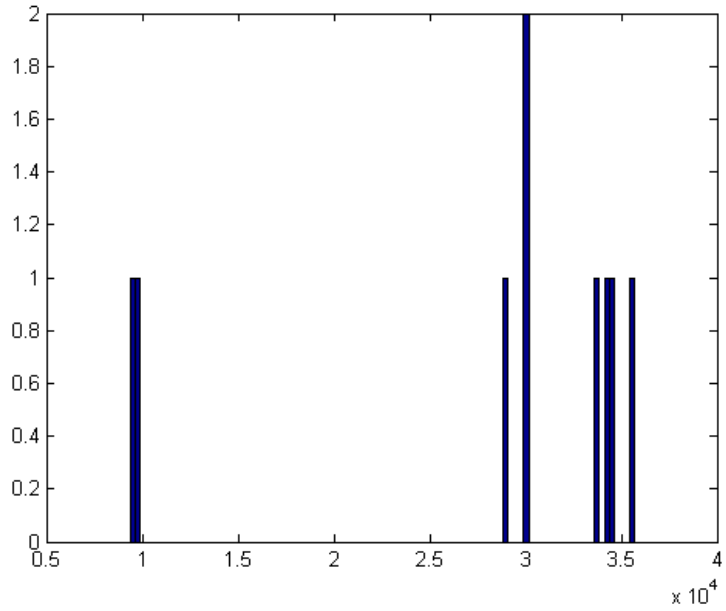

Figure 3.5: Three clusters were found by the k-means method. Again, it is shown that only a single instance of a structure needs to be found in L-SAC.

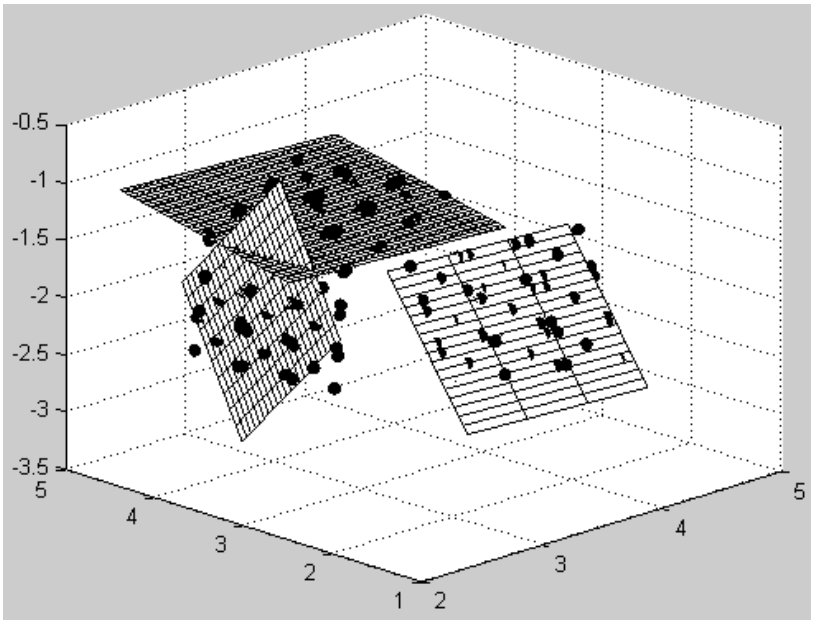

Figure 3.6: Discovered planes plotted over back-projected points.

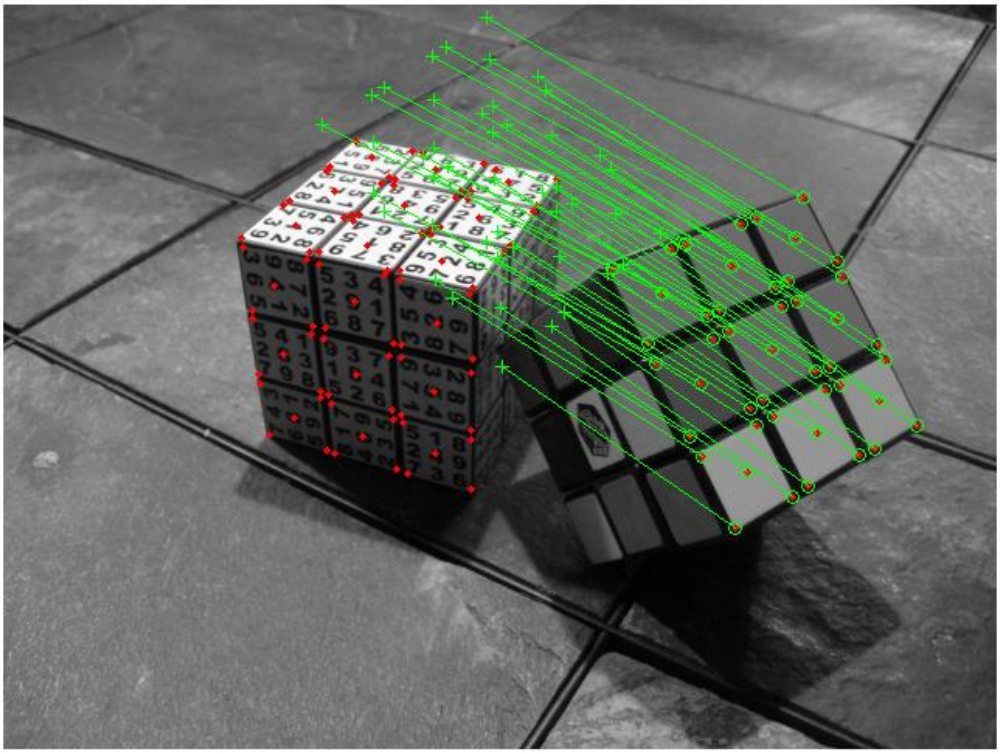

Figure 3.7: Example of points being projected to another plane using the recovered planes to find a homography.

4 EIGENFACE STRUCTURES

4.1 Application to Finding Human Faces in Xbox Kinect Data

Finding multiple faces using the Xbox Kinect was attempted using the L-SAC method for finding multiple structures. Images of multiple scenes involving faces were captured using the Kinect sensor. One of the greatest difficulties to searching for multiple faces in an image is the complex nature of the face. A simple linear or even simple non-linear function cannot fully describe a face. This fact means that it will take many data points to adequately describe a face so the dataset to search through will be very large. How can such a complex shape be described by a MSS and a relatively small yet unique descriptor vector that is suitable for distinguishing consensus sets? To accomplish this, the method of eigenfaces by Turk *et al.* was used to recognize whether the data presented in the image contained faces

Eigenfaces was introduced in [17] as a method to perform facial recognition from a database of known faces. Eigenfaces could also learn new faces by recognizing an unknown image as a face, then adding it to the database of known faces. It was discovered by Kirby and Sirovich that in principle any face could be rebuilt using only a small collection of weights for each face and a set of standard pictures. Performing Principal Component Analysis (PCA) on the covariance matrix of the set of face images produces the eigenvectors which store the variation between each image. Only the M best eigenvectors corresponding to the M most significant

eigenvalues are needed. These M eigenvectors are called eigenfaces, and any face image can be represented by a linear combination of these eigenfaces.

To calculate the eigenfaces, first the image of size N is converted from a 2D array to a vector $\mathbf{\Theta}_i$ of length N^2. The database of face images is then the set, $\{\mathbf{\Theta}_1, \mathbf{\Theta}_2, \dots, \mathbf{\Theta}_M\}$. The mean of the database is computed as,

$$\mathbf{X} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{\Theta}_i \,. \tag{4.1}$$

The difference of each face vector to the average is $\mathbf{\Phi_i} = \mathbf{\Theta}_i - \mathbf{X}$.

PCA is used to find the eigenvectors $\mathbf{u}_k$ and eigenvalues λ_k. The covariance matrix is,

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{\Phi}_i \mathbf{\Phi}_i^{\mathrm{T}} = \mathbf{A}\,\mathbf{A}^{\mathrm{T}}, \tag{4.2}$$

where $\mathbf{A} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2, \dots, \mathbf{\Phi}_M]$.

In order to reduce the number of calculations needed, the eigenvectors $\mathbf{v}_i$ of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ give,

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{v_i} = \mu\mathbf{v_i}. \tag{4.3}$$

If both sides are pre-multiplied by A, then,

$$\mathbf{A}\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{v}_i = \mu\mathbf{A}\mathbf{v}_i, \tag{4.4}$$

which shows that $\mathbf{A}\mathbf{v}_i$ are the eigenvectors of $\mathbf{C} = \mathbf{A}\mathbf{A}^{\mathrm{T}}$. The MxM matrix $\mathbf{L} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$, where $L_{mn} = \mathbf{\Phi}_m^{\mathrm{T}}\mathbf{\Phi}_n$ is formed. The eigenvectors $\mathbf{v}_i$ of $\mathbf{L}$ determine the linear combinations of the database face images. The eigenfaces $\mathbf{u}_l$ are thus,

$$\mathbf{u}_l = \sum_{k=1}^{M} \mathbf{v}_{lk} \mathbf{\Phi}_k, \quad l = 1,2, \dots M. \tag{4.5}$$

A face is transformed into "face space" by,

$$\omega_k = \mathbf{u}_k^{\mathrm{T}} (\mathbf{\Theta} - \mathbf{X}), \qquad k = 1,2, \dots M, \tag{4.6}$$

and the collection of the weights is the vector $\mathbf{\Omega} = [\omega_1, \omega_2, \dots, \omega_M]^{\mathrm{T}}$.

Now that the eigenfaces of the database are found, and there is a method to transform each image into "face space," a method can be created to determine faces in an unknown image. A region in the unknown image the size of a face image is selected, and transformed into "face space," using $\mathbf{\Phi}_f = \sum_{i=1}^{M} \omega_i \mathbf{u}_i$, where $\mathbf{\Phi}_f$ is the "face space" vector. The difference of the unknown image is taken with the mean face as, $\mathbf{\Phi} = \mathbf{\Theta} - \mathbf{X}$. The squared error between $\mathbf{\Phi}$ and $\mathbf{\Phi}_f$ is

$$\epsilon^2 = \left\| \mathbf{\Phi} - \mathbf{\Phi}_f \right\|^2, \tag{4.7}$$

and if this error is below a defined threshold t, the unknown image is considered a face.

The Kinect provides an RGB image and a depth image where each pixel value is the z-coordinate real world depth value in millimeters. Xu *et al.* [19] extended the idea of eigenfaces for 3D mesh models built from 3D point cloud data. The premise is simply to treat each (x,y,z) coordinate as a pixel, where x and y are the grid, and the z value serves as the intensity value. Each mesh model built from point cloud data was normalized so that the vertices of the meshes aligned across the database. This is an important step since the images in the database need to be the same size and aligned.

For the experiments in this thesis, the depth image produced by the Kinect was used to search for faces. For the procedure in this paper, the depth image was converted to real world (x,y,z) coordinates and point cloud data extracted. However mesh models were not produced from these point clouds, but the point cloud data was used directly.

In order to use the depth image, the x and y world coordinate values needed to be calculated. Rather than perform a calibration procedure on the Kinect, the x-y mapping for the depth pixels was done using the values from [6],

```
fx_d = 1.0 / 5.9421434211923247e+02;
fy_d = 1.0 / 5.9104053696870778e+02;
cx_d = 3.3930780975300314e+02;
cy_d = 2.4273913761751615e+02;
```
(4.8)

where fx_d, fy_d are the focal length parameters and cx_d, cy_d are the principle point parameters. The depth pixels were then converted to real world (x,y,z) coordinates by,

$$\begin{aligned} z &= -z, \\ x &= (\mathrm{col} - \mathrm{cx\_d})(z)(\mathrm{fx\_d}), \\ y &= (\mathrm{row} - \mathrm{cy\_d})(z)(\mathrm{fy\_d}). \end{aligned} \tag{4.9}$$

With the depth image mapped to real world coordinates, a 3D point cloud of the scene could be constructed and mapped to a 2D image.

The next step involved building a small database by extracting face images from the collected depth images. Six faces were captured to form a database. Each image is a 2D m x n matrix,

$$\mathbf{\Phi}_i = \begin{bmatrix} z_{1,1} & z_{1,2} \dots & z_{1,n} \\ \vdots & \vdots & \vdots \\ z_{m,1} & z_{m,2} \dots & z_{m,n} \end{bmatrix}, \tag{4.10}$$

where m and n are odd values, and z is the real world z-coordinate represented as the intensity value of the pixel. All faces are aligned by placing the nose tip in the center of the matrix. The eigenfaces are computed using the process above to yield the matrix

$$\mathbf{U} = [\mathbf{u}_1\, \mathbf{u}_2\, \dots\, \mathbf{u}_k\,], \tag{4.11}$$

where $\mathbf{u}_i$ is the eigenface corresponding to the i^{th} eigenvalue. In this experiment, $k = 6$, and only the 5 most significant eigenfaces were used. The Kinect sensor captures significant noise around features such as hair, glasses, and edges, therefore the database images appear quite noisy to the eye. However, only two pre-processing steps were performed on the image. The depth images were scaled to increase the dynamic range between pixel depth, and a depth threshold was applied to remove the background. The eigenface database can be seen in Figure 4.1 and Figure 4.2 below.

Figure 4.1: RGB Images of the database.

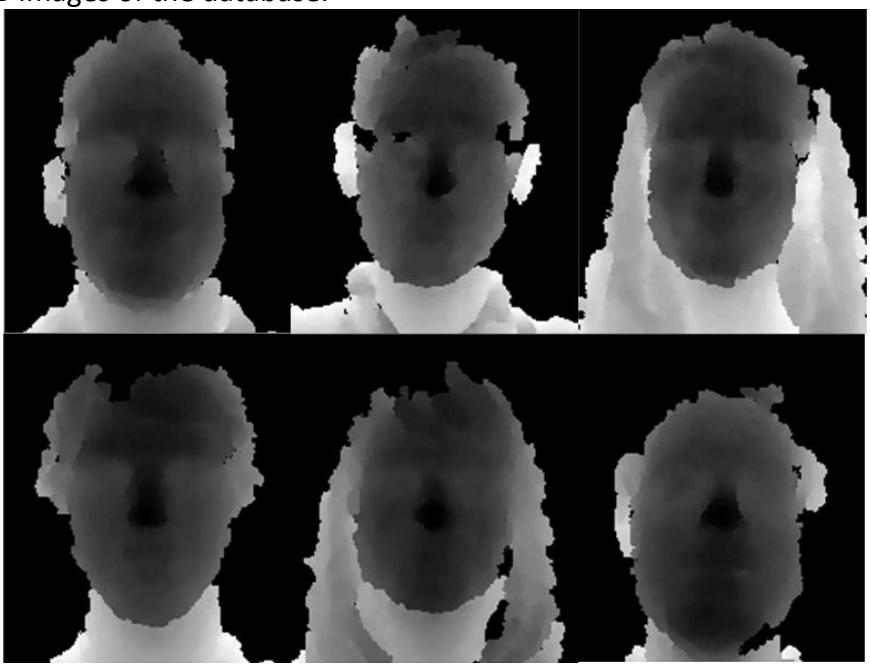

Figure 4.2: Depth Images of the database

With the eigenface database built, it is important to define a MSS and descriptor vector so that the random sampling procedure can be performed. A 3D face has been defined above as a 2D m x n matrix where m and n are odd. Therefore the MSS can be represented as simply:

$$z_{m/2,n/2}. \tag{4.12}$$

Only one point needs to be sampled and $\mathbf{\Phi}$ will be captured form this center point.

Such a simple MSS leads to a simple descriptor vector, as the coordinates of this center point are descriptive enough to define a unique CS. If the sampled dataset is the image $\mathbf{I}$ of size i x j, then the descriptor vector $\mathbf{D}$ is

$$\mathbf{D} = [i\ j]^{\mathrm{T}}. \tag{4.13}$$

As the points in the image are sampled and $\mathbf{I}$ is projected onto "face space", the cost function is evaluated. If the i^{th} image is below t, it is considered a potential face, and $\mathbf{D_i}$ is formed from the (i,j) coordinates. The threshold t is determined heuristically. The norm of this vector is taken and stored forming the set

$$S_d = \{d_1, d_2, \dots, d_k\}, \tag{4.14}$$

where k is the number of potential faces found.

Once S_d is obtained, the number of faces can be determined. The values of d relating to the i^{th} face will tend to form a distinct cluster if $\mathbf{D}$ is a good descriptor, so a simple *k*-means clustering procedure will work well as an unsupervised method to organize the data and extract the number of faces existing in the image. For *k*-means to work properly, it is important for the initial estimate of means to be sufficiently far apart. The initial guess for the number of cluster

means will be the estimated maximum number of potential faces in the image determined by the size of $\mathbf{I}$ divided by the size of $\mathbf{\Phi}$ rounded down to the nearest integer. To ensure that the initial means are sufficiently far apart, the initial mean is place at d_{min}. It is important that an initial mean estimate does not get placed in-between two actual clusters, otherwise a false cluster will form and the correct number of faces cannot be extracted. Therefore it is important to define a step size such that means do not overlap. If the size of $\mathbf{\Phi}$ is $m\ x\ n$ then if a step size for the remaining means is defined as:

$$\delta \stackrel{\text{def}}{=} \sqrt{m^2 + n^2}, \tag{4.15}$$

then it is ensured each mean cannot overlap two faces. With the initial means formed, the points d are associated to their closet mean, and the mean of each new set is formed. This process iterates until no changes are made. The final clusters are of the faces in the image. The CS of each cluster is found using a least-squares fit. This is a simple procedure since the cost function is a least-squares fit already. Then the $\epsilon_{min}^2$of each cluster found when evaluating the cost function gives the CS and the best fit face is represented from this CS.

4.2 Examples Using Xbox Kinect

Three experiments were performed using the Xbox Kinect to test the effectiveness of L-SAC. The first example involved extracting 3 different faces existing in the eigenface database. The second example included two faces, one in the database and one completely unknown

face. The final example includes a face in the database, and an image of a face in the database to show that L-SAC will only recover the actual face. The data is presented below.

The first experiment involved three faces from the database. Each face in the database was a 170 x 146 array so the minimum detectable model size is 24,820. The input image to the system was 240 x 550 meaning the dataset to sample was 132,000 points. Three faces in the image mean there are 57,540 or 43.6% gross outliers. Only 4000 samples without replacement were taken or 3.03% of points. All 3 faces were found in 10.08 seconds in MATLAB®. The input image is below.

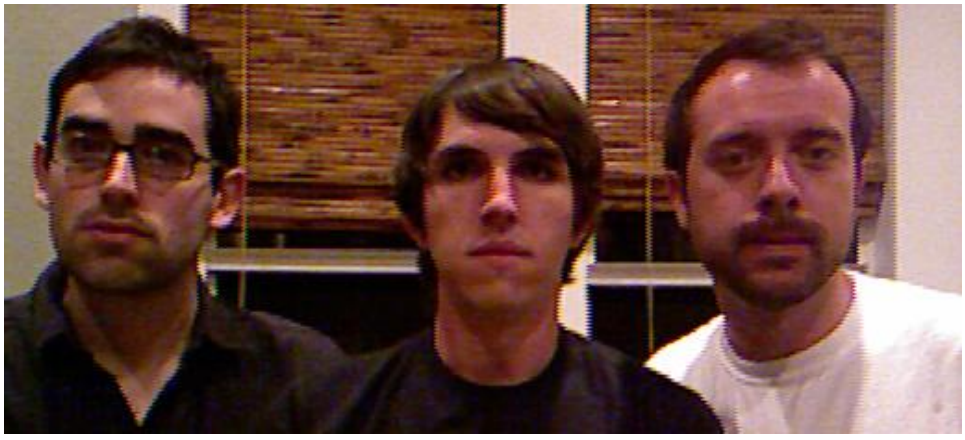

Figure 4.3: Three faces from eigenface database (RGB)

Figure 4.4: Structure candidates found during random sampling (Depth Image). The image is a 16-bit image, and the depth values have been scaled to maximize the dynamic range of 0-65535.

Figure 4.5 below is a histogram of the clusters formed by taking the norm of the descriptor vector $\mathbf{D}$ of each face candidate.

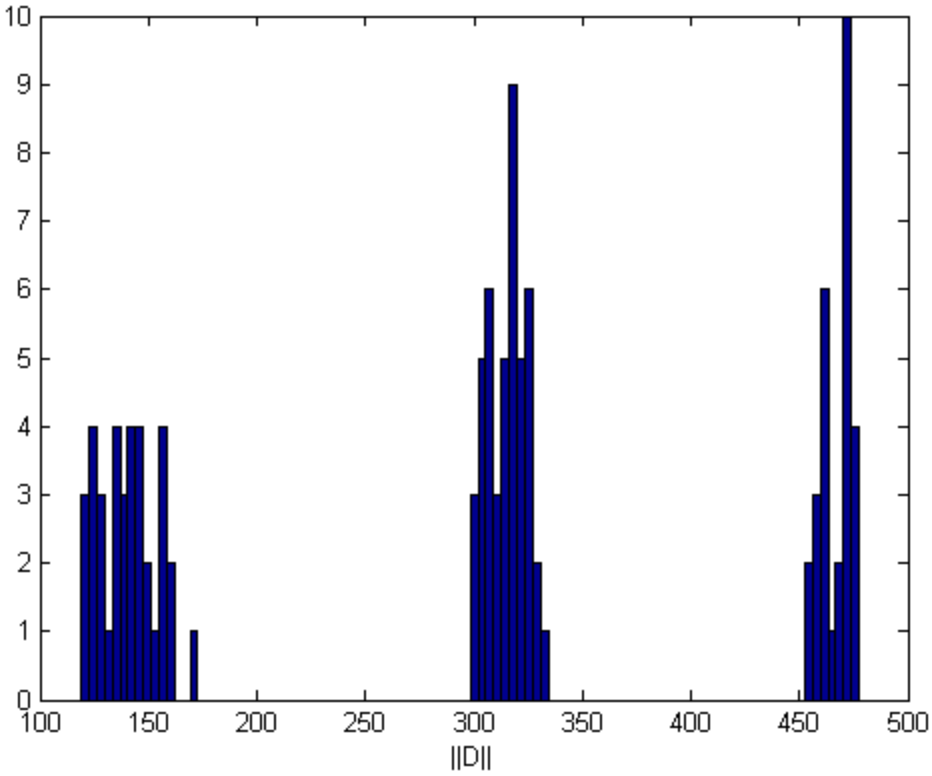

Figure 4.5: The norm of the descriptor vector of each face candidate found during the sampling procedure. The descriptor vector D produces very distinct peaks around each actual face. A simple k-means clustering can extract the unique face data.

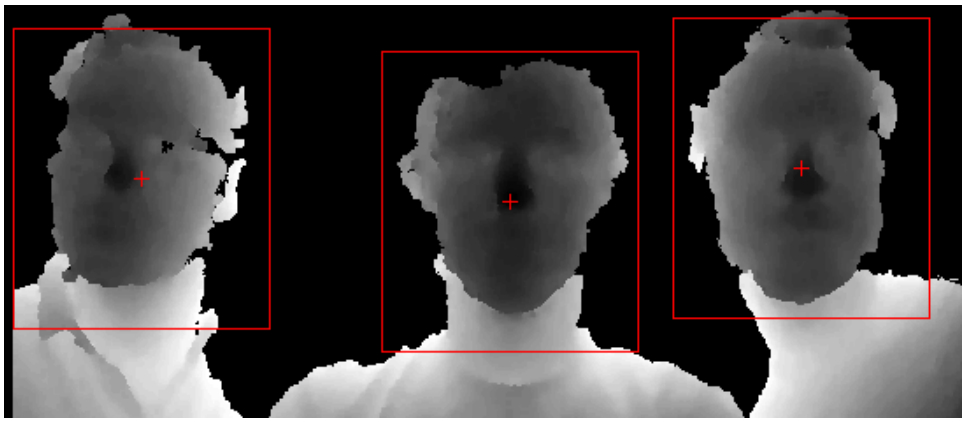

Figure 4.6: Final fit of faces found in the input image. The best fit of each face was found by performing a least-squares fit of each cluster.

The next experiment included an “unknown” face not found in the database. The input image was 280 x 550, while the face was again 170 x 146. Two faces in the image means the image contained 67.78% gross outliers. Again only 4000 samples were taken or 2.6% of the points in the dataset. Both faces were found in 13.70 seconds.

Figure 4.7: Two faces, one from the database and one "unknown" face. (RGB)

Figure 4.8: Face candidates found during sampling process.

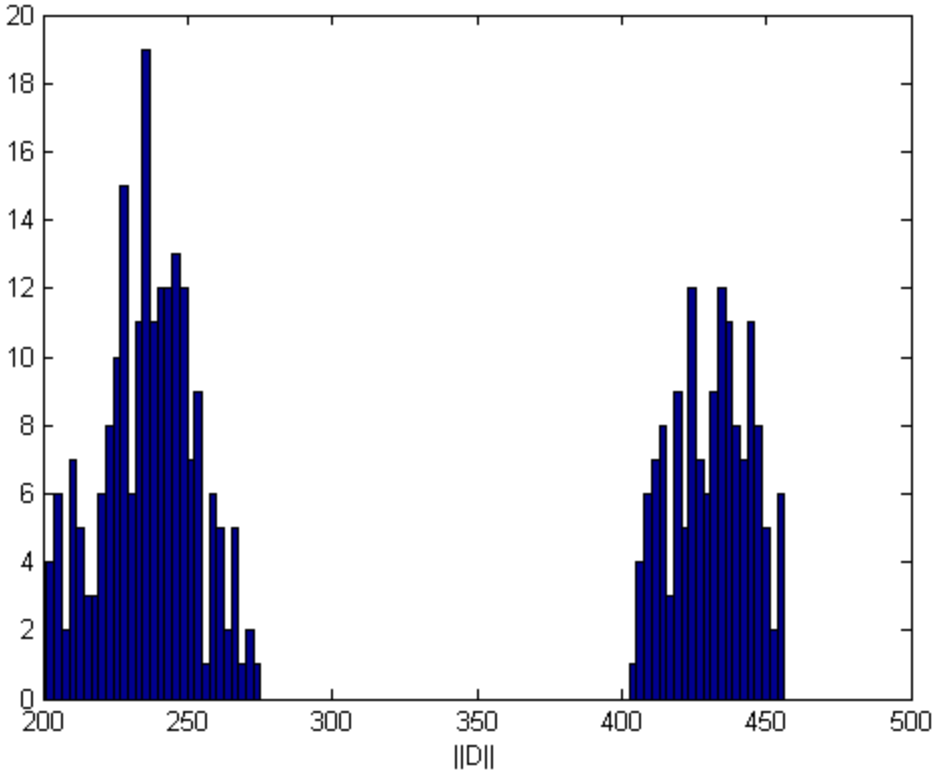

Figure 4.9: Again, two very distinct clusters generated by the norm of D.

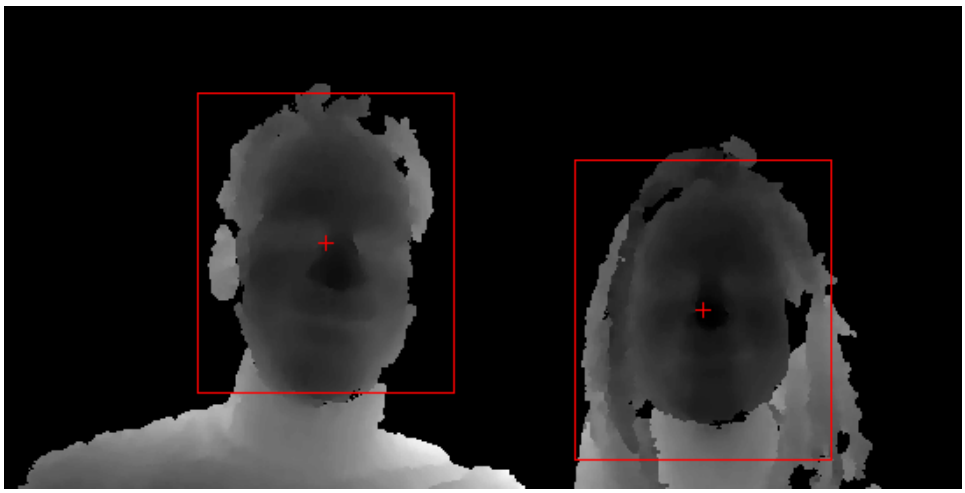

Figure 4.10: Final least-squares fit of each face. As can be seen above, even faces not contained in the database can be discovered in the image.

The final experiment was a demonstration to show that if depth images are used, an image of a face will not fool the algorithm. The input image contained a face from the database, and a life-size image of the same face. The input image size was 280 x 550, and the face size of 170 x 146. The number of gross outliers was 83.88% and the number of samples taken was 4000. The face was found in 12.55 seconds.

Figure 4.11: Input image of a face from the database and an image of that face. (RGB)

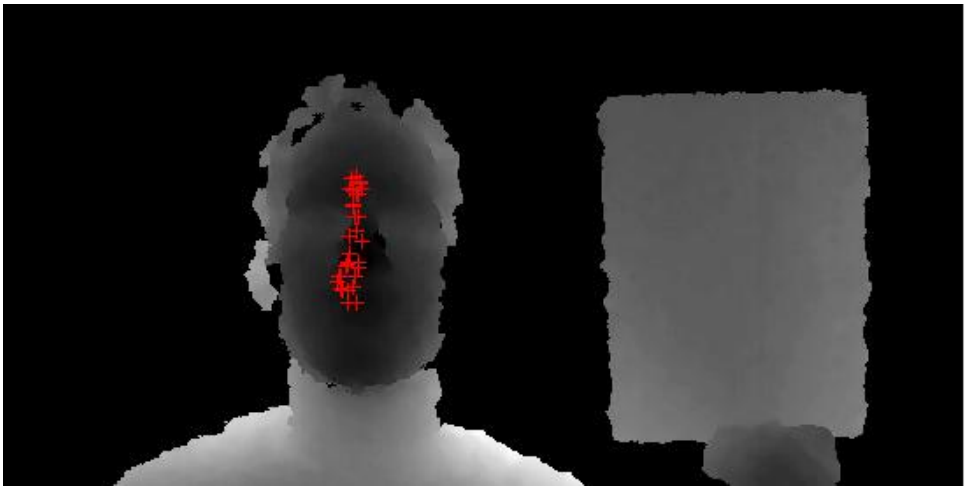

Figure 4.12: Face candidates found during sampling process.

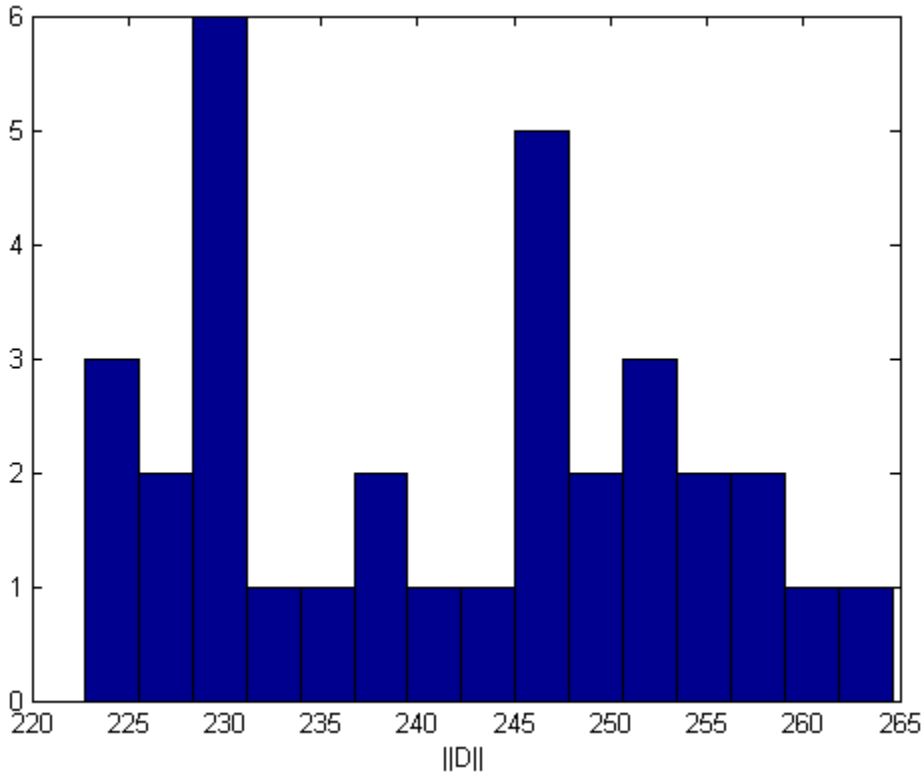

Figure 4.13: Histogram of the norms of each face candidates descriptor vector.

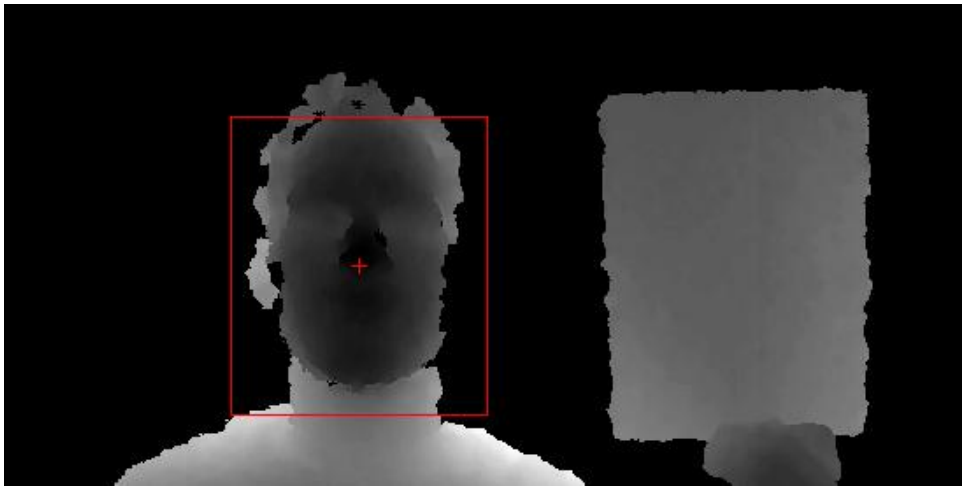

Figure 4.14: Final fit of the single face as the imaged face does not contain depth variation and therefore is not detected as a face.

5 CONCLUSION

5.1 Future Development

There are two areas of improvement to L-SAC that were noted while developing the algorithm. Currently a *k*-means clustering algorithm is used and was chosen for its simplicity, time constraints, and its effectiveness for datasets where the density of structures were low to intermediate. As the structure density increases, *k*-means' ability to correctly differentiate unique instances degrades as shown in Figure 5.1. This is because the initial set of means is unknown and estimated, and k-means is highly variable depending on the initial mean selection. It is easily seen that the boundary conditions in Figure 5.1 are still very distinctive, therefore the inability to correctly cluster is not due to a weakness in the L-SAC framework, but a weakness in the clustering. With more time, a better clustering scheme can be developed to overcome this problem as the number of structures in a dataset increase. It should still be noted that only J-Linkage demonstrated their algorithm with a similar number of structures in a dataset. Even with *k*-means clustering, L-SAC performed to a similar level as J-Linkage in this regard as shown in Figure 5.2.

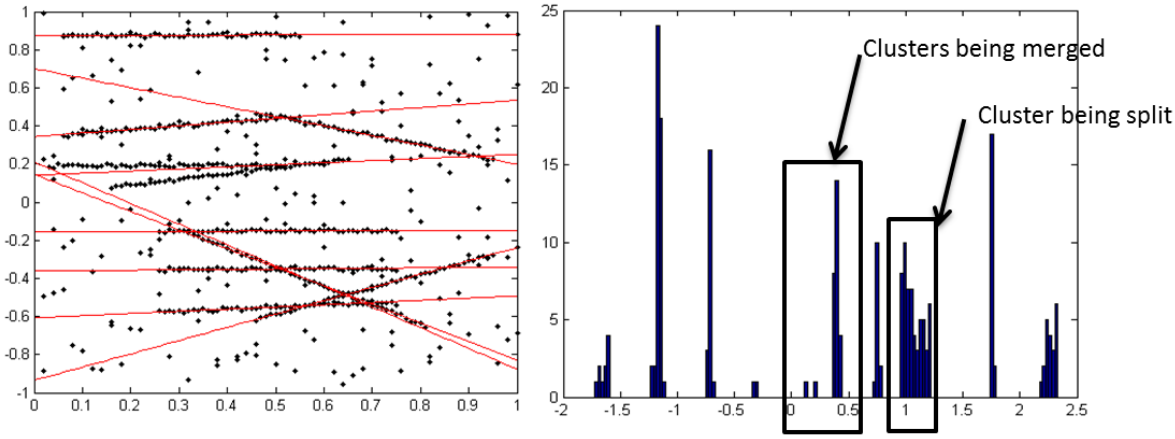

Figure 5.1: It can be seen in the fitted line picture that 2 lines were merged together in one instance, while one line was split into 2 in another. This is because *k*-means initialization caused incorrect clustering as can be seen from the histogram. Ten distinct peaks are shown but because *k*-means is highly variable on the initial estimation, the clusters were incorrectly categorized. A more sophisticated technique would work better here. Again it can be seen that only 2 instances of a line are more than enough to recover it.

The second area of improvement noticed was that it may be possible to not instantiate all combinations from the random sample set. For instance, if 5 points on a single structure have been sampled, all combinations of these five points do not need to be used since L-SAC can recover the structure with only one instantiation. This improvement could significantly reduce computational complexity for structures with a large MSS.

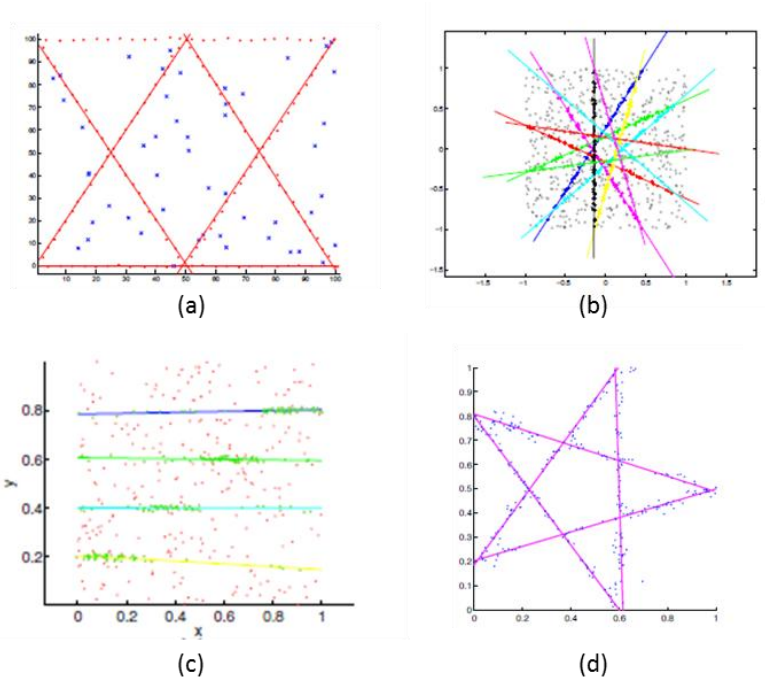

Figure 5.2: Line fitting demonstrations from the literature. Figure (a) is from RHA [23], (b) is from J-Linkage [15], (c) is from multiRANSAC [25], and (d) comes from KF [14]. It can be seen that only J-Linkage demonstrated their algorithm with significant structures, 11 total. Even with a naïve k-means clustering, L-SAC performs at a similar level, without needing to know the number of points per structure like J-Linkage.

5.2 Conclusions and Discussion

RANSAC's popularity and robustness for a single fitting structure in noisy datasets provided motivation to extend the capabilities of sampling consensus algorithms to apply in a multi-structure environment. The current state of the art in this field was examined and presented in this thesis. Then a novel approach was called Labeled Sampling Consensus (L-SAC) was introduced and applied to the trivial case of finding lines in a noisy dataset. L-SAC was

shown to be highly effective even in situations where the number of structures in the dataset was relatively large.

Due to the fact that L-SAC seeks a label that is highly unique and amplifies the differences in similar structures, the clusters formed had very distinctive boundaries. By relying on boundary conditions rather than analyzing the modes in the distribution, it was shown that a structure can be found even if only one instance was instantiated from the random sample set. This ability of L-SAC allowed for the development of a novel sampling technique, dramatically "compressing" the number of samples needed, and guaranteeing all structures existing can be found.

Finally, L-SAC was shown to be a powerful and flexible framework that can be generalized to a diverse set of problem domains. In this thesis it was applied to discovering planes and homographies in a stereo-vision environment, and discovering human faces using point-cloud data obtained from the Xbox Kinect. In both environments, L-SAC proved effective in finding all structures with a compressed sample set. In conclusion, the work done in this thesis provides a very promising starting point to an effective method that can be implemented in real life systems.

REFERENCES

[1] Abramowitz, M. and Stegun, I. A., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Ninth ed. N.Y., USA: Dover Publications, Inc., 1972.

[2] Comaniciu, D., and Meer, P., "Mean Shift Analysis and Applications," in *International Conference on Computer Vision (ICCV)*, 1999, pp. 1197-1203.

[3] Duda, R. O. and Hart, P. E., "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11-15, Jan. 1972.

[4] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*, 2nd ed. N.Y., USA: John Wiley & Sons Inc., 2001.

[5] Fischler M. A., and Bolles, R.C., "Random Sample Consesnsus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, pp. 381-395, June 1981.

[6] Matthew Fisher. (2011, May) Matt's Webcorner. [Online]. http://graphics.stanford.edu/~mdfisher/Kinect.html

[7] Hartley, R., and Zisserman, A., *Multiple View Geometry*, 2nd ed. N.Y., USA: Cambridge University Press, 2003.

[8] Kanazawa, Y., and Kawakami, H., "Detection of Planar Regions With Uncalibrated Stereo Using Distributions of Feature Points," in *British Machine Vision Conference*, 2004, pp. 247-256.

[9] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*, 3rd ed. Upper Saddle River, USA: Pearson Prentice Hall, 2008.

[10] J.B., MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Proceedings of 5th Berkeley Symposium on Methematical Statistics and Probability*, Berkeley, 1967, pp. 281-297.

[11] Rousseeuw, P. J. and Leroy, A. M., *Robust Regression and Outlier Detection*.: Wiley, 1987.

[12] Shawne-Taylor, J., and Cristianini, *Kernel Methods for Pattern Analysis*. N.Y.,: Cambridge University Press, 2004.

[13] N. J. A. Sloane. (2011, May) The On-line Encyclopedia of Integer Sequences. [Online]. http://oeis.org/A000041

[14] Tat-Jun Chin, Hanzi Wang, and Suter, D., "Robust Fitting of Multiple Structures: The Statistical Learning Approach," in *IEEE 12th International Conference on Computer Vision*, Kyoto, Jp, 2009, pp. 413-420.

[15] Toldo, R., and Fusiello, A., "Robust Multiple Structures Estimation with J-Linkage," in *Computer Vision - ECCV 2008*. Marseille, France: Springer-Verlag Berlin Heidelberg, 2008, pp. 537-547.

[16] Torr, P. and Zisserman, A., "MLESAC: A New Robust Estimator with Applications to Estimating Image Geometry.," *Computer Vision and Image Understanding (CVIU)*, vol. 78, no. 1, pp. 138-156, April 2000.

[17] Turk, M., and Pentland, A., "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[18] Wang, H. and Suter, D., "MDPE: A Very Robust Estimator for Model Fitting and Range Image Segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 139-166, Sept. 2004.

[19] Xu, C., Wang, Y., Tan, T., and Quan, L., "A New Attempt to Face Recognition Using 3D Eigenfaces," in *Asian Conference on Computer Vision (ACCV)*, 2004, pp. 884-889.

[20] Xu, L., Oja, E., and Kultanen, P., "A New Curve Detection Method: Randomized Hough Transform (RHT)," *Pattern Recognition Letters*, vol. 11, no. 5, pp. 331-338, May 1990.

[21] Zhengyou Zhang, "A Flexible New Technique for Camera Calibration," Microsoft Research, Redmond, WA, MSR-TR-98-71 1998.

[22] Zhengyou Zhang. (April, 2009) A Flexible New Technique for Camera Calibration. [Online]. http://research.microsoft.com/en-us/um/people/zhang/Calib/

[23] Zhang, W., and Kosecka, J., "Nonparametric Estimation of Multiple Structures with Outliers," in *Dynamical Vision, ICCV 2005 and ECCV 2006 Workshops*.: Springer-Verlag Berlin Heidelberg, 2007, pp. 60-74.

[24] Zoghbi, A., and Stojmenovic, I., "Fast Algorithms for Generating Integer Partions," *International Journal of Computer Math*, vol. 70, pp. 319-332, 1998.

[25] Zuliani M., Kenney C.S., and Manjunath B.S., "The MultiRANSAC Algorithm and Its Application to Detect Planar Homographies," in *International Conference on Image Processing (ICIP)*, 2005, pp. III-153-6.