

2009

Spatially Coherent RANSAC for Multi-Model Fitting

Hossam N. Isack
Western University

Follow this and additional works at: <https://ir.lib.uwo.ca/digitizedtheses>

Recommended Citation

Isack, Hossam N., "Spatially Coherent RANSAC for Multi-Model Fitting" (2009). *Digitized Theses*. 4261.
<https://ir.lib.uwo.ca/digitizedtheses/4261>

This Thesis is brought to you for free and open access by the Digitized Special Collections at Scholarship@Western. It has been accepted for inclusion in Digitized Theses by an authorized administrator of Scholarship@Western. For more information, please contact wlsadmin@uwo.ca.

Spatially Coherent RANSAC for Multi-Model Fitting

(Spine Title: Spatially Coherent RANSAC for Multi-Model Fitting)

(Thesis Format: Monograph)

by

Hossam N. Isack

Graduate Program in Computer Science

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

2

School of Graduate and Postdoctoral Studies

University of Western Ontario

London, Ontario

April 2009

© Hossam N. Isack 2009

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES

CERTIFICATE OF EXAMINATION

Supervisor

Examiners

Yuri Boykov

Steven Beauchemin

Supervisory Committee

Eric Schost

Roy Eagleson

The thesis by

Hossam N. Isack

entitled

Spatially Coherent RANSAC for Multi-Model Fitting

is accepted in partial fulfillment of the
requirements for the degree of

Master of Science

Date

Chair of Thesis Examination Board

Abstract

RANSAC [15, 38, 1] is a reliable method for fitting parametric models to sparse data with many outliers. Originally designed for extracting a single model, RANSAC also has variants for fitting multiple models when supported by data. Our main insight is that, in practice, inliers for each model are often spatially coherent — all previous RANSAC-based methods ignore this. Our new method fits an unspecified number of models to data by combining ideas of random sampling and spatial regularization. As in basic RANSAC, we randomly sample data points to generate a set of proposed models (labels). We formulate model selection and inlier classification as a single problem — labeling of triangulated data points. Geometric fit errors and spatial coherence are combined in one MRF-based energy. In contrast to basic RANSAC, inlier classification does not depend on a fixed threshold. Moreover, our optimization framework allows iterative re-estimation of models/inliers with a clear stopping criteria and convergence guarantees. We show that our new method, SCORANSAC, can significantly improve results on synthetic and real data supporting multiple linear, affine, and homographic models.

Keywords: RANSAC, α -expansion, multi-model fitting, discrete optimization, graph cut algorithms, energy minimization.

Acknowledgements

To Professor Yuri, thank you for your patience and for investing your time in me. I hope that one day I will be able to repay you. I apologize for my silence. I hope to find the right time to speak one day soon.

“If it is not truthful and not helpful, don’t say it. If it is truthful and not helpful, don’t say it. If it is not truthful and helpful, don’t say it. If it is truthful and helpful, wait for the right time.”

– Buddha

To Olga Veksler, thank you for providing the α -expansion code, which is the soul of SCORANSAC, and thanks for making sense out of my stammering when no one else could.

To Andrew, I know we did not speak or share much, but I always felt that you will be there if I needed you.

“The person I like the most is the one who points out my defects.”

– El-Amir Umar (Radia Allahu Anhu)

I would also like to extend my gratitude to David Lowe for providing the compiled binaries of the SIFT keypoint detector.

If any of the original authors of the Figures 1.3, 2.3(b), 3.1, 3.6(c), 3.13(a-b), 3.14(a-b), 3.15(a-b), 3.15(d), 3.16(a-c), 3.2.3.3(a-b), 3.18(a-b), and 3.19 want to remove his figures, contact the author of this publication.

Contents

Certificate of Examination	ii
Abstract	iii
Acknowledgements	iv
List of Figures	viii
List of Tables	xiv
List of Algorithms	xv
1 Introduction	1
1.1 Model Fitting in Computer Vision	1
1.2 Single Model Fitting	2
1.2.1 Basic RANSAC	2
1.2.2 Applications	6
1.3 Multi-Model Fitting	7
1.3.1 Sequential RANSAC	7
1.3.2 MultiRANSAC	9
1.3.3 Applications	12
1.3.4 Shortcomings of Standard RANSAC-Based Methods	13
1.3.4.1 Sensitivity to Threshold Parameter τ	13
1.3.4.2 Ambiguous Points	14
1.4 From Thresholding to Regularization	17

1.4.1	Regularization and MRF-based models in Computer Vision	18
1.4.2	Optimization and α -expansion	19
1.5	Our Contribution	20
2	SCO-RANSAC	21
2.1	Overview of the Main Idea	22
2.2	Initial Labels \mathcal{L}_0	22
2.2.1	Outlier Labels	23
2.3	Energy Formulation	25
2.4	Inlier Classification and Model Re-estimation	26
2.5	Our Algorithm	29
2.6	General Properties	30
2.6.1	Noise and Outliers	30
2.6.2	Number of Initial Samples	35
2.6.3	Accuracy of Estimated Models	36
2.6.4	Performance	37
2.6.5	Overcoming the Limitations of Previous Methods	38
2.7	Merging Spatially Overlapping Models	40
3	Applications in Computer Vision	42
3.1	Fitting Multiple Affine Models	43
3.1.1	Estimating Planes in Narrow-Based Stereo	43
3.1.2	Geometric Fit Measures	45
3.1.2.1	Horizontal Shift	45
3.1.2.2	Weighted Shift	48
3.1.3	SCO-RANSAC vs. Birchfield and Tomasi	52
3.2	Fitting Multiple Homographies	57
3.2.1	Estimating Planes in Uncalibrated Wide-Based Stereo	57
3.2.2	Removing Outliers and Retriangulating Holes	57
3.2.3	Comparisons and Results	59
3.2.3.1	SCO-RANSAC vs. Sequential RANSAC Methods	62

3.2.3.2	SCO-RANSAC vs. Multi-RANSAC	64
3.2.3.3	Scenes with Large Number of Models	65
3.3	Conclusion	69
3.4	Extensions	69
A	Plane-to-Plane Transformation in Rectified Images	70
B	Levenberg-Marquard	73
B.1	Minimizing The Symmetric Transfer Error Using LM	75
C	Outlier Gap in the Case of a 2D Line	77
	Bibliography	80
	Vita	85

List of Figures

1.1	Figure (a) shows a data set containing 80/80 inliers/outliers which was perturbed by a Gaussian noise ($\sigma = 0.0075$). Figure (b) shows each model generated by randomly sampling a MSS (two points shown as red 'X'). Figure (c) shows the consensus set of the model, which consists of all the points that lie inside the shaded area. Figure (d) shows the model with the largest CS.	3
1.2	Figure (a) shows a data set that contains 80/80 inliers/outliers which was perturbed by a Gaussian noise ($\sigma = 0.0075$). The outliers were added to form a cluster. Figure (b) shows that RANSAC failed to find the right model.	5
1.3	The results of image mosaicing algorithms that use RANSAC as a robust estimation algorithm.	6
1.4	Figure (a) show the initial data set of 80 inliers per model and 160 outliers, perturbed by a Gaussian noise ($\sigma = 0.0075$). Figure (b) shows the model found in the first iteration. Figure (c) shows reduced data set, which is the initial data set after removing the inliers of the first model. Figure (d) shows the model found by applying basic RANSAC algorithm to the reduced data set.	8
1.5	Figure (a) shows the data set contains 3 models, 40 inliers per model. Two models were perturbed with a low level of Gaussian noise and the other one with a higher level noise. MultiRANSAC results for using a low threshold and a high threshold are shown in Figures (b) and (c) respectively.	14

1.6	Competing models, Figure (a) shows two models with spatially separated sets of inliers. Figure (b) show for the horizontal line/model in which the blue area represents the set of all points that are at distance τ or less from the line/model. Any point that lies within that area fits the horizontal line. The same is true for the vertical line and the green area. The red circle depicts the ambiguous set.	15
1.7	Figure (a) shows the data set contains two models, 40 inliers each. These two models are competing models. Both models were perturbed with the same Gaussian noise. Figure (b) shows the result of the ambiguous set being assigned to the correct model. Figure (c) shows the result of the ambiguous set being assigned to the wrong model.	16
1.8	17
2.1	Figure (a) shows a data set with 150 inliers and one outlier (shown in red). Figure (b) shows ten initial proposals. As you can see its unlikely for the outlier to get sampled unless the number of initial proposals was really large. To avoid this we introduce an outlier label for each data point.	24
2.2	Illustrations of SCO-RANSAC's iterations. Figure (a) show the proposals generated by random sampling. Figures (b) through (f) show the re-estimation of models and their inliers. This is an illustrative example. SCO-RANSAC only produces a reasonable result with a certain probability. This probability increases as more initial proposals are used.	27
2.3	Delaunay triangulations of data points. Other techniques can be used, particularly for higher dimensional data.	28
2.4	The seven Original models used in generating the synthetic data sets.	30
2.5	Comparing the results for fitting lines to noisy data points. The data points were perturbed with a low level of Gaussian noise ($\sigma = 0.005$) and 120 outliers were added.	31

2.6	A Comparison of the results for fitting lines to noisy data points. The data points were perturbed with a medium level of Gaussian noise ($\sigma = 0.01$) and 200 outliers were added.	32
2.7	A comparison of the results for fitting lines to noisy data points. The data points were perturbed with a high level of Gaussian noise ($\sigma = 0.02$) and 300 outliers were added.	34
2.8	The results for using SCO-RANSAC for fitting lines to noisy data points. The data points were perturbed with a very high level of Gaussian noise ($\sigma = 0.025$) and 500 outliers were added.	35
2.9	Error vs. the number of randomly sampled models plot for RANSAC and SCO-RANSAC which are shown in blue and red, respectively	36
2.10	This figure shows SCO-RANSAC results for the data sets presented in Figure 1.2(a), Figure 1.7(a) and Figure 1.5(a), respectively. These results demonstrate that SCO-RANSAC was able to overcome the pitfalls of the RANSAC-based approaches, and the clustered noise situation in which basic RANSAC failed.	39
2.11	This figure illustrates the case of spatially overlapping models. Figure (a) shows the data set. Figures (b) and (c) show the two results that SCO-RANSAC generated for different runs. Figure (d) shows the guaranteed result after adding the post processing step.	41
3.1	Figures (a) and (b) show the SIFT matches found on the left and right images, respectively. Figure (c) shows the matched features between the two images.	44
3.2	This figure illustrates the horizontal shift error d_{hr} . The distance between two triangulated points is denoted by d . The first triangulated point is obtained from p_l and p_r (shown in red). The second triangulated point is obtained from p_l and the mapping of p_l onto the second image by A_π . A_π is the affine/model induced by the plane π . The image of d onto the second image is denoted by d_{hr}	46

3.3	Figures (a) and (b) show the original stereo pairs obtained by BT [4]. Figures (c) and (d) show SCO-RANSAC result for using the horizontal shift error (3.3) and the weighted shift error (3.6).	47
3.4	That figure illustrates that the horizontal shift error 3.3 does not take the slope of a plane into account. Obviously, the red point is closer to plane π_2 , even though the horizontal shift error d_{hr} is the same for both π_1 and π_2	49
3.5	This figure illustrates the effect of allowing the epipolar line of matched pair of points p_l, p_r to shift. Figure (a) shows that line L belongs to the plane π and it is mapped as l and l' on the left and right images, respectively. Figure (b) shows the 2D view of the two rectified image planes.	51
3.6	Comparisons of the results for the Clorox stereo pair [4]. Figures (a) and (b) show sparse inlier classification by SCO-RANSAC using different geometric fit measures. Figure (c) show dense pixel segmentation by BT [4] using photoconsistency. BT's method did not segment the Clorox box correctly, and it split the plane corresponding to the books.	53
3.7	We computed the ground truth affine models for the planes A_1, \dots, A_4 using 10 manually matched features (shown in white) and a line L_4 . The lines L_1, \dots, L_4 (shown in green) are the lines of intersection computed using the ground truth affine models. The computed affine model for the ground plane was up to 0.25 pixel accurate.	54
3.8	SCO-RANSAC using the horizontal shift error. Lines were computed for all pairs of intersecting planes (affine models).	55
3.9	BT [4] using photoconsistency measure. Lines were computed for all pairs of intersecting planes (affine models).	56
3.10	SCO-RANSAC using the weighted shift error. Lines were computed for all pairs of intersecting planes (affine models).	56
3.11	This figure shows the gap that an outlier for the plane π will introduce if points are triangulated in one of the image planes instead of 3D.	58

3.12	Figure (a) shows the general case of an isolated outlier, in which we only know that the yellow edges were severed. Figure (b) the isolated outlier was removed and the hole was retriangulated. The new edges are shown as dotted lines (see appendix C for further details on the effect of this operation on the energy).	59
3.13	SCO-RANSAC was able to find three planes. The original images are taken from Marc Pollefeys's Leuven castle data set.	60
3.14	SCO-RANSAC was able to find two planes. The original images are from VGG (Oxford) Merton College. SCO-RANSAC was able to accurately classify the inliers. Notice the place where the two planes intersect.	61
3.15	This figure compares Sequential RANSAC and Kanazawa and Kawakami method to SCO-RANSAC. The original stereo pair si from Kanazawa and Kawakami [24].	63
3.16	Comparing multiRANSAC and SCO-RANSAC using stereo images from VGG (Oxford) Merton College I.	64
3.17	Stereo images from VGG (Oxford) Merton College III.	66
3.18	In this example from VGG (Oxford), we used the same color more than once to represent different planes. Only spatially connected planes are shown in different colors. Figure (a) shows red and green rectangles, which are drawn to facilitate their referencing in the text.	67
3.19	Image of Raglan Castle tower found on Flickr. Anonymous photographer. .	68
C.1	This figure shows a line fitting example for the 2D data points/2D triangulation case (first column) and the 2D data points/1D triangulation case (second column). The first row shows the data sets, the second row shows the triangulation, and the third row shows the SCO-RANSAC results. The severed edges are marked by a cross. In Figures (b) and (d), the outlier is shown in red to help the reader track it, but in Figure (f), the red data point is an isolated data point and represents a separate model.	78

C.2	Figure (a) shows the result of removing the outlier (an isolated data point) and retriangulating the hole. By applying the next SCO-RANSAC iteration on the new graph, we obtain the result shown in Figure (b). This iteration makes use of the new spatial connectivity information and merges the two models. Retriangulating the hole (caused by the outlier) makes the merging feasible.	79
-----	---	----

List of Tables

1.1	Notations used in describing the multiRANSAC algorithm.	9
2.1	Convergence, this table shows the number of labels \tilde{N} remaining after each iteration for four independent runs of SCO-RANSAC using the data set shown in Figure 2.7(a). Notice how the first iteration eliminated most of the hypotheses.	38
3.1	Show the geometric errors for lines where affine models intersect. Errors were computed as the sum of distances between the ground truth line segment corners and the computed lines.	55

List of Algorithms

1	Basic RANSAC	5
2	Sequential RANSAC	8
3	multiRANSAC	11
4	Greedy Merging Algorithm of MultiRANSAC	11
5	SCO-RANSAC	29

Chapter 1

Introduction

1.1 Model Fitting in Computer Vision

Model fitting is widely used in data analysis. The problem of model estimation becomes more complex when the noise level in the data increases and when it is corrupted by outliers. There are two main approaches to model fitting in such cases. The first approach is “robust regression” [32], which tries to generate model estimates that are not affected by outliers by randomly sampling a large number of model hypotheses. Least Median Square (LMS) and Random Sample Consensus (RANSAC) [15] are the most common robust regression techniques used in computer vision.

The second approach is called “regression diagnostics” [32]. These techniques compute outlier diagnostics, which are statistics that are used to detect data points that have a significant influence on the estimator. Regression diagnostics are also used in computer vision. For example, Wei Zhang and Jana Kosecka [44] estimate multiple models using a technique that follows a regression diagnostics paradigm. They detect and estimate multiple models in the data by analyzing the distribution of the errors for individual data points with respect to a set of hypotheses. The hypotheses are generated by a RANSAC-like sampling process.

Our contribution is a novel general multi-model fitting method, Spatially Coherent RANSAC (SCO-RANSAC). Our method follows the robust regression paradigm. The novelty is incorporation of spatial coherency of the inliers for each model. In a nutshell, SCO-RANSAC searches for a set of spatially coherent models that fits any given data set. It works by generating a set of model hypotheses at random, and then iterating over identi-

fitting their spatial support and refitting the models.

Our main assumption is that inliers for all models are spatially coherent. Our insight is that this assumption is valid in many applications particularly in computer vision. Yet, sometimes the set of inliers of a model may have spatial gaps either due to an overlap with another model or due to missing data or a large number of outliers. In sections 2.7 and 3.2.2, we show how SCO-RANSAC deals with these problems.

Our method does not require the number of models to be specified. It automatically finds the least number of models that fit the data. In particular it may hallucinate some models among outliers [see Fig. 2.2(f)] such weak models are easy to prune out using a simple post processing step. SCO-RANSAC could be extended by adding a diagnostic step to filter out the weak models, as there are no constraints on the number of models that SCO-RANSAC can find. In that step, one could possibly use a diagnostic technique or criteria to filter out weak models (e.g. based on the goodness of fit), such an extension is further explored in section 3.2.2.

In the following sections, we will discuss existing single and multi-model fitting techniques. In the context of single model fitting, we will consider the RANSAC [15] algorithm. We will also discuss Sequential RANSAC [39] and multiRANSAC [45] as the RANSAC extensions for fitting multiple models.

1.2 Single Model Fitting

1.2.1 Basic RANSAC

The RANSAC algorithm (1) proposed by Fischler and Bolles [15], is a general model-fitting approach designed to deal with a large number of outliers in the data. We will continue our discussion of RANSAC and RANSAC-based methods in the context of synthetic multi-line fitting examples in 2D [see Fig. 1.1(a)].

Definition 1 *The Minimal Sample Set (MSS) is a set that contains the minimal number of data points required to estimate a model. For example, when fitting line models each minimal sample set consists of two points.*

Definition 2 The consensus set (CS) is the set of data points that fits a given model, according to a given error measurement metric and a predefined threshold τ [see Fig. 1.1(c)].

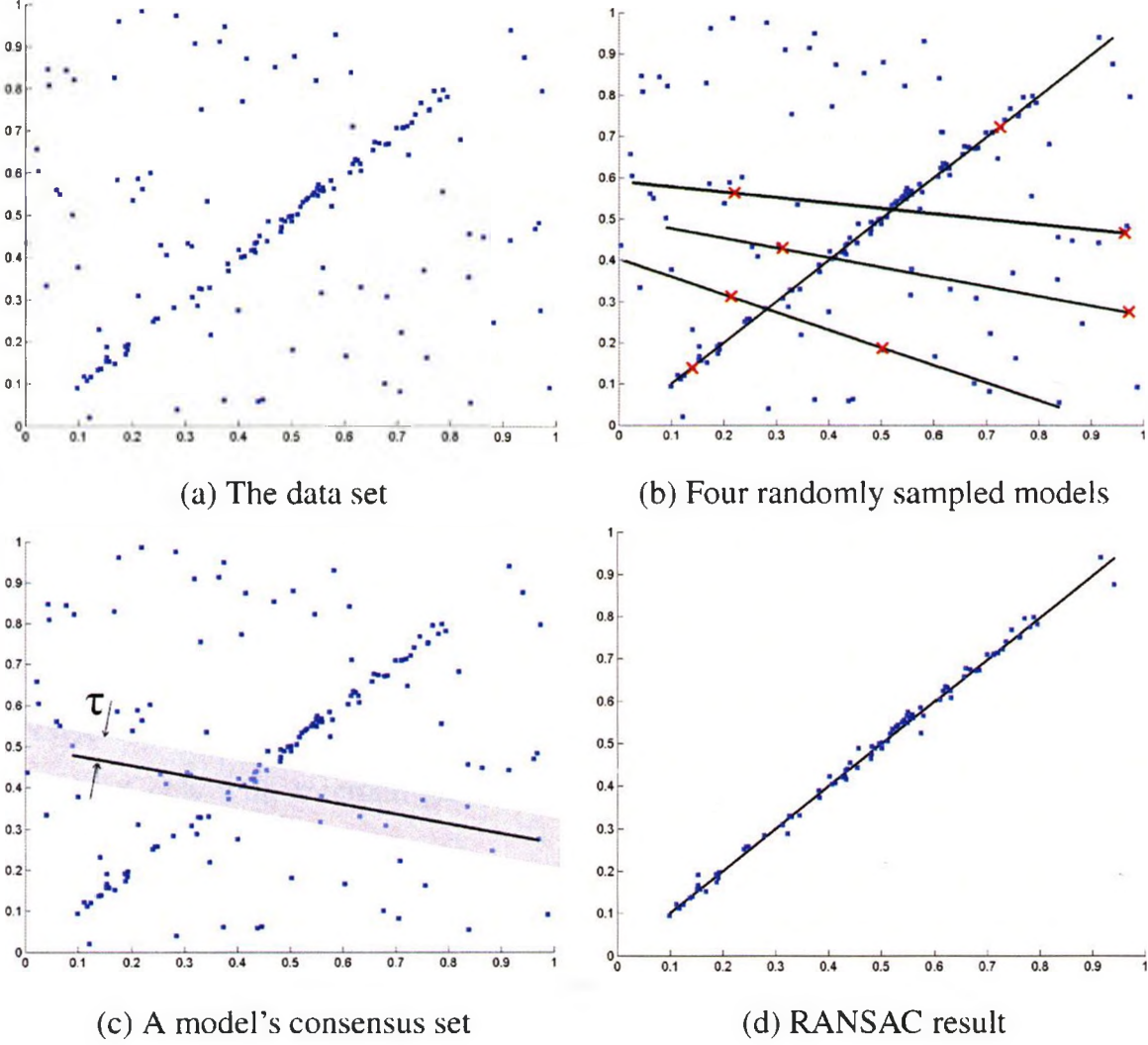


Figure 1.1: Figure (a) shows a data set containing 80/80 inliers/outliers which was perturbed by a Gaussian noise ($\sigma = 0.0075$). Figure (b) shows each model generated by randomly sampling a MSS (two points shown as red 'X'). Figure (c) shows the consensus set of the model, which consists of all the points that lie inside the shaded area. Figure (d) shows the model with the largest CS.

RANSAC iteratively samples a random minimal sample set to estimate a model [see Fig. 1.1(b)]. It then finds the model's consensus set [see Fig. 1.1(c)]. This process is repeated a certain number of times while keeping only the largest consensus set [see Fig.

1.1(d)]. Finally, the model parameters are re-estimated using the largest consensus set. The rationale for selecting a model with the largest consensus set is that if one of the points of the minimal sample set is an outlier then the model estimated from this minimal sample set will not gain much support, the result being a small consensus set.

In each iteration, RANSAC samples a random minimal sample set and estimates a model from that set. The number of iterations required by RANSAC to guarantee a global optimal solution is often computationally infeasible, as that would be equivalent to exhaustive search over all possible minimal sample sets. Instead, a sufficiently large number of iterations is chosen to ensure with probability k , that at least one of the sampled minimal sample sets is free from outliers. Suppose t is the probability that any selected data point is an inlier, while assuming that the cardinality of the minimal sample set is n . Then at least N iterations are required so that $(1 - t^n)^N = 1 - k$ [19] thus,

$$N = \frac{\log(1 - k)}{\log(1 - t^n)} \quad (1.1)$$

RANSAC and many RANSAC-based algorithms are nondeterministic as they produce a good model only with a certain probability, which increases as the number of iterations (randomly sampled models) increases.

The major advantage of RANSAC is that it can estimate the model parameters with high accuracy even when the data contains a large proportion of outliers and is perturbed by a high level of noise. One of the disadvantages of RANSAC is its sensitivity to the threshold parameter τ . Also, it can only fit one model to a given data set. When there exists more than one model, it may fail to find any of them. Wang and Suter [41] showed that RANSAC is more likely to fail in cases in which there are clustered outliers (see Fig. 1.2). One possible way to overcome this problem is to estimate multiple models and then select the best model using a criteria other than the number of inliers.

RANSAC is more accurate and efficient than the Randomized Hough Transform (RHT). Both RHT and RANSAC reduce the search space by random sampling. RHT keeps track of the votes for the sampled models by discretizing the model space, it then selects the model with the largest number of votes as the estimated model. As you can see the dis-

cretization of the model space could lead to less accurate models, and the discretizing high dimensional spaces will not be efficient.

Algorithm 1 Basic RANSAC

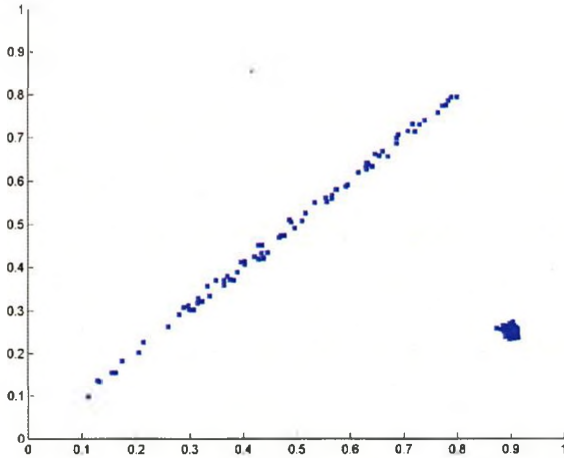
input:

- P – The data set of points
- n – Minimum number of points required to estimate the model parameters
- k – The probability that at least one of the sampled MSSs is free from outliers
- t – The probability that any selected data point is an inlier
- τ – The threshold for determining whether a data point fits a model or not

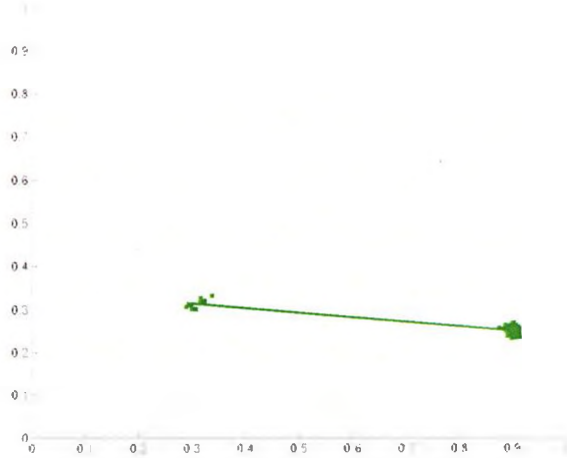
output :

- m – Estimated model parameters
 - s – The CS of the found model
-

- 1: $N = \frac{\log(1-k)}{\log(1-t^n)}$
 - 2: **for** $i = 0$ to N **do**
 - 3: $m \leftarrow$ Select n points at random and solve for the model parameters.
 - 4: $s^* \leftarrow$ Find the CS of the model m according to the threshold τ .
 - 5: $s \leftarrow$ Select the largest set from s and s^* .
 - 6: **end for**
 - 7: $m \leftarrow$ Estimate the model parameters using the largest CS found, s .
-



(a) The data set



(b) RANSAC result

Figure 1.2: Figure (a) shows a data set that contains 80/80 inliers/outliers which was perturbed by a Gaussian noise ($\sigma = 0.0075$). The outliers were added to form a cluster. Figure (b) shows that RANSAC failed to find the right model.

1.2.2 Applications

In computer vision, RANSAC is widely used for estimating homographies (e.g. in mosaicing), fundamental matrices [38], trifocal tensors [1], and for many other problems [19]. RANSAC was also used by Le Lu et al. [29] to implement a robust 3D face tracking technique. Capel and Zisserman [10] used RANSAC to simultaneously estimate a homography and its consensus set for image mosaicing [see Fig. 1.3(a)]. Brown and Lowe [8, 9] used RANSAC in creating their fully automatic panorama construction technique [see Fig. 1.3(b)]. In computer vision, many problems are based on fitting a set of data points to some suitable model.



(a) The Keble mosaic projected onto a cylinder,
Original image from Capel and Zisserman [10].



(b) Recognized panorama number 1,
Original image from Brown and Lowe [8].

Figure 1.3: The results of image mosaicing algorithms that use RANSAC as a robust estimation algorithm.

1.3 Multi-Model Fitting

The Standard RANSAC algorithm is restricted to cases where a single model accounts for all of the inliers in the data. Sequential RANSAC [39] and multiRANSAC [45] were proposed to generalize the RANSAC algorithm to the case when data supports multiple models.

1.3.1 Sequential RANSAC

The Sequential RANSAC algorithm (2) is the simplest and the most straightforward generalization of the basic RANSAC algorithm. Given a set of data points that supports multiple models, the Sequential RANSAC algorithm finds the first model by applying basic RANSAC algorithm. The next model is found in the same way after the inliers of the first model are removed from the data (see Fig. 1.4). Sequential RANSAC continues in the same manner until there are no more models with enough data support.

The main disadvantage of the Sequential RANSAC approach is its high sensitivity towards the threshold parameter. Sequential RANSAC is more sensitive to the threshold parameter because it uses the same threshold parameter for all models. For example, Sequential RANSAC fails in case each model contains different noise level, since different threshold values are necessary to identify inliers for each model. Using a single threshold value for all the models will lead to inaccurate inlier detection, which in turn will affect the overall robustness. As mentioned in [42], reducing the data sequentially by the removing the inliers of the dominant model as in [42] could prevent the detection of other models. This issue is more critical in case of motion detection due to the phantom motion, for more information the reader is referred to [37].

Algorithm 2 Sequential RANSAC

input:
 p, n, k, t and τ – These are the basic RANSAC parameters

output :
 M – The set of found models which best fit the data

 S – The set of corresponding CSs of the found models

```

1: repeat
2:    $(m, s) \leftarrow \text{Basic RANSAC}(P, n, k, t, \tau)$ 
3:    $M = M \cup \{m\}$ 
4:    $S = S \cup \{s\}$ 
5:    $P = P - s$ 
6: until no more models can be found.
  
```

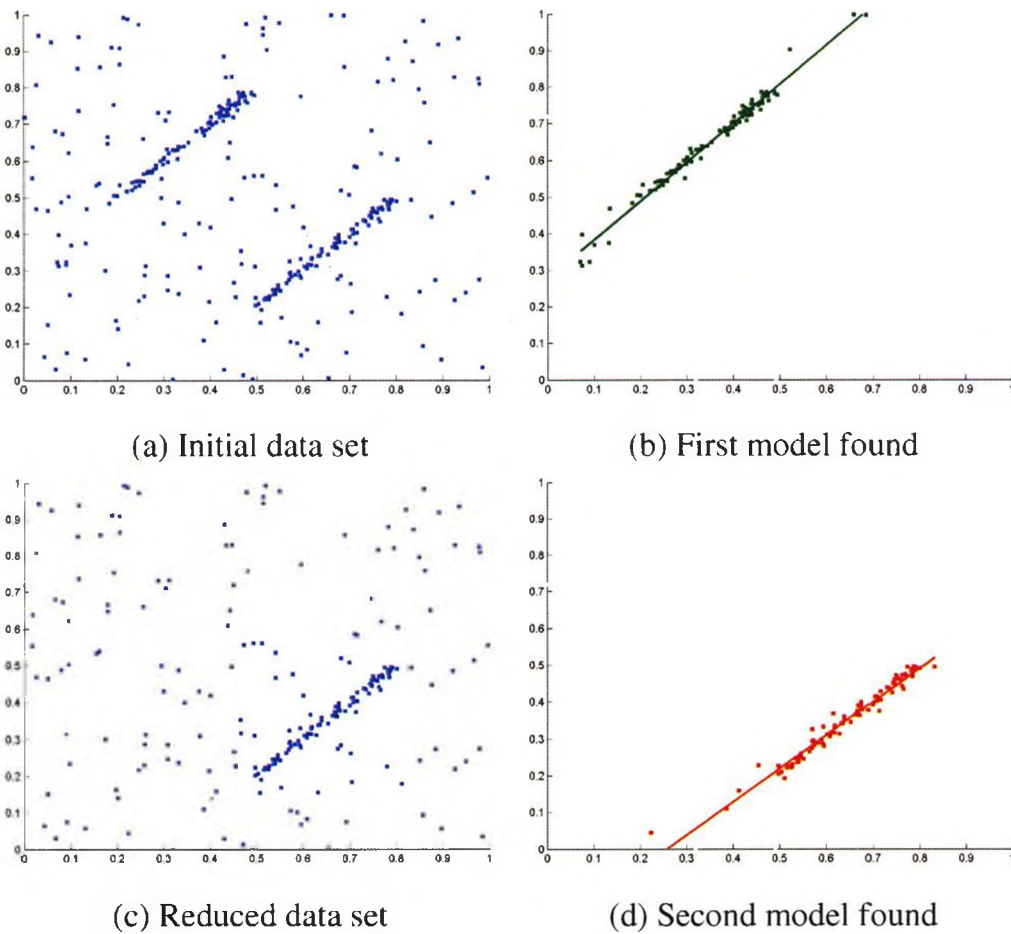


Figure 1.4: Figure (a) show the initial data set of 80 inliers per model and 160 outliers, perturbed by a Gaussian noise ($\sigma = 0.0075$). Figure (b) shows the model found in the first iteration. Figure (c) shows reduced data set, which is the initial data set after removing the inliers of the first model. Figure (d) shows the model found by applying basic RANSAC algorithm to the reduced data set.

1.3.2 MultiRANSAC

We will use the notations shown in Table 1.1 to describe the multiRANSAC algorithm [45]. The multiRANSAC algorithm (3) requires prior knowledge of the number of models to be estimated, W . Assuming that the data set supports W distinct models, the multiRANSAC algorithm works as follows. At each iteration, multiRANSAC randomly draws a minimal sample set and computes its corresponding consensus set. It then removes the found consensus set from the data set. The same procedure is repeated $(W - 1)$ times to find the remaining models. The probability of sampling W minimal sample sets composed entirely of inliers, given that the number of inliers for the W models are $|P_1|, |P_2|, \dots, |P_W|$, is

$$q = \frac{\binom{|P_1|}{n} \binom{|P_2|}{n} \dots \binom{|P_W|}{n}}{\binom{|P|}{n} \binom{|P|-|P_1|}{n} \dots \binom{|P|-\sum_{w=1}^{W-1} |P_w|}{n}}. \quad (1.2)$$

Notation	Description
P	The given data set of points
P_i	The set of inliers of the model i
$ V $	Denotes the cardinality of the set V
W	The number of models
n	The minimum number of data points required to define a model
s_i	MSS of the model i
S_i	The corresponding CS of the MSS s_i
S	The set of CSs of the W models
M	The set of W models
N	Is the number of iterations required by multiRANSAC
η	Probability that multiRANSAC fails

Table 1.1: Notations used in describing the multiRANSAC algorithm.

The probability of having one or more minimal sample set with members other than those from the inliers sets is $(1 - q)$. When randomly sampling W minimal sample sets N times, the probability that no iteration will contain W minimal sample sets that are composed entirely of inliers is given by $(1 - q)^N$, where

$$\lim_{N \rightarrow \infty} (1 - q)^N = 0.$$

The value of N should be large enough to ensure that probability $(1 - q)^N$ is less than some specified value η :

$$N = \frac{\log \eta}{\log 1 - q}. \quad (1.3)$$

It is not possible to estimate the number of iterations required by the multiRANSAC algorithm using (1.3), since the number of inliers of each model is not known. Under the assumption that the size of the consensus set of a model is smaller than or equal to the number of inliers of that model, it is possible to approximate the value of N for every iteration by using the cardinalities of consensus sets found in the previous iteration:

$$\hat{q}(S) = \frac{\binom{|S_1|}{n} \binom{|S_2|}{n} \dots \binom{|S_W|}{n}}{\binom{|P|}{n} \binom{|P|-|S_1|}{n} \dots \binom{|P|-\sum_{w=1}^{W-1} |S_w|}{n}}$$

$$\hat{N}(S) = \frac{\log \eta}{\log 1 - \hat{q}(S)}. \quad (1.4)$$

In summary, in each iteration multiRANSAC randomly samples W models, and removes the consensus set from the original data set of a model once found. The number of iterations required by the multiRANSAC can be approximated using (1.4). The consensus sets found in each iteration are merged with those of the previous iteration, keeping only the largest W disjoint consensus sets. The merging algorithm (4) is greedy, as it combines $2W$ consensus sets of two consecutive iterations to find W consensus sets that maximize the total number of inliers.

Algorithm 3 multiRANSAC

input:

- P – The set of data points
- n – Minimum number of points required to estimate the model parameters
- W – Number of models
- η – Probability that multiRANSAC fails
- τ – A threshold for determining when a data point fits a model

output :

- M – the set of found models which best fit the data
 - S – the set of corresponding CSs of the found models
-

```

1:  $S \leftarrow \phi$ 
2:  $\hat{N} \leftarrow \infty$ 
3: for  $i = 0$  to  $\hat{N}$  do
4:    $V \leftarrow \phi$ ;  $i \leftarrow i + 1$ ;  $P^* \leftarrow D$ 
5:   for  $w = 1$  to  $W$  do
6:      $m \leftarrow$  Sample a random MSS and compute the corresponding model  $m$ 
7:      $S^* \leftarrow$  Find the CS of the model  $m$ 
8:      $P^* \leftarrow P^* - S^*$ 
9:      $V \leftarrow V \cup S^*$ 
10:  end for
11:   $S \leftarrow$  Greedily merge  $S$  and  $V$ 
12:   $\hat{N} \leftarrow \frac{\log \eta}{\log 1 - \hat{q}(S)}$ 
13: end for
14:  $M \leftarrow$  Compute the models of the final  $W$  CSs in  $S$ 

```

Algorithm 4 Greedy Merging Algorithm of MultiRANSAC

input:

- V – The set of W CSs at iteration i
- T – The set of W CSs at iteration $i + 1$
- W – The number of models

output :

- S – The merged W CSs of the models
-

```

1:  $S_{all} \leftarrow V \cup T$ 
2:  $S \leftarrow \phi$ 
3: repeat
4:    $S^* \leftarrow$  CS with the largest cardinality from  $S_{all}$ 
5:   if  $S^* \cap S = \phi$  then
6:      $S \leftarrow S \cup S^*$ 
7:      $w \leftarrow w + 1$ 
8:   end if
9:    $S_{all} \leftarrow S_{all} - S^*$ 
10: until  $w = W$ 

```

1.3.3 Applications

Zhang and Kosecka [44] proposed a diagnostic approach to identify multiple models and outliers by analyzing the distribution of the residuals of individual data points with respect to the set of hypotheses generated by a RANSAC-like sampling process. They also used their technique in motion segmentation for just two models of foreground/background motions.

Vincent and Laganier [39] introduced Sequential RANSAC algorithm in the context of detecting multiple homographies in an image pair. Kanazawa and Kawakami [24] detected local planar regions in a scene by estimating multiple homographies using a variant of Sequential RANSAC. The Kanazawa and Kawakami technique will be covered in greater detail in section 3.2.3.1.

Zuliani et al. [45] introduced multiRANSAC algorithm to overcome some of the limitations of the Sequential RANSAC algorithm [39]. They used multiRANSAC in estimating multiple lines and multiple homographies. López-Nicolás et al. [27] proposed a new method for estimating multiple homographies from two views of a multi-plane scene using matched lines and points. Later, they estimated multiple homographies to compute the fundamental matrix between two views [33]. Qiang He and Chee-hung [21] assumed known camera calibration for estimating multiple *planar homographies*. This was done by enforcing homographic constraints, as proposed by Faugeras and Luong [14].

We believe that practical applicability of existing multi-model fitting methods is severely restricted by their quality, see section 1.3.4 and experimental comparisons in section 2.6.1. The goal of this work is overcome the limitations of the current techniques. Our approach also suggest further extensions to piece-wise smooth fitting of models, see section 3.3.

1.3.4 Shortcomings of Standard RANSAC-Based Methods

The RANSAC-based approaches covered in the previous sections suffer from two major problems. The first one is their high sensitivity to the threshold parameter, especially if models have different noise levels. In the cases in which the data set contains a subset of points that can fit into more than one model, we refer to these models as competing models and to the subset of points as the ambiguous set of points. Standard RANSAC-based approaches are random when assigning the ambiguous set of points to one of the competing models. This section will elaborate on these disadvantages. Only the multiRANSAC approach will be considered in this discussion.

1.3.4.1 Sensitivity to Threshold Parameter τ

The multiRANSAC algorithm uses the same threshold value in finding the consensus sets for all models. This only works if the inliers of each model are perturbed at the same noise level. The data set shown in Figure 1.5(a) shows three models with 40 inliers each and no outliers, for simplicity. For two of the models, the inliers were perturbed with a low level of Gaussian noise. The parameters of these two models are very close; thus, low threshold value is required for correct inlier classification for each model. The third model was perturbed with a high level of Gaussian noise, which requires large threshold to enable multiRANSAC to identify its inliers correctly.

The result of using a low threshold is shown in Figure 1.5(b). The two models with the low noise levels were correctly identified, while most of the inliers of the model with the high noise level were marked as outliers. Figure 1.5(c) shows the result for a higher threshold. The CS of the model with the high noise level was identified correctly, but the two models with the low noise level were identified as one. This example also shows another disadvantage of the multiRANSAC approach. Since it requires a priori knowledge of the number of models W , it fits exactly W models to the data. This could be a problem if data supports a number of models different from W .

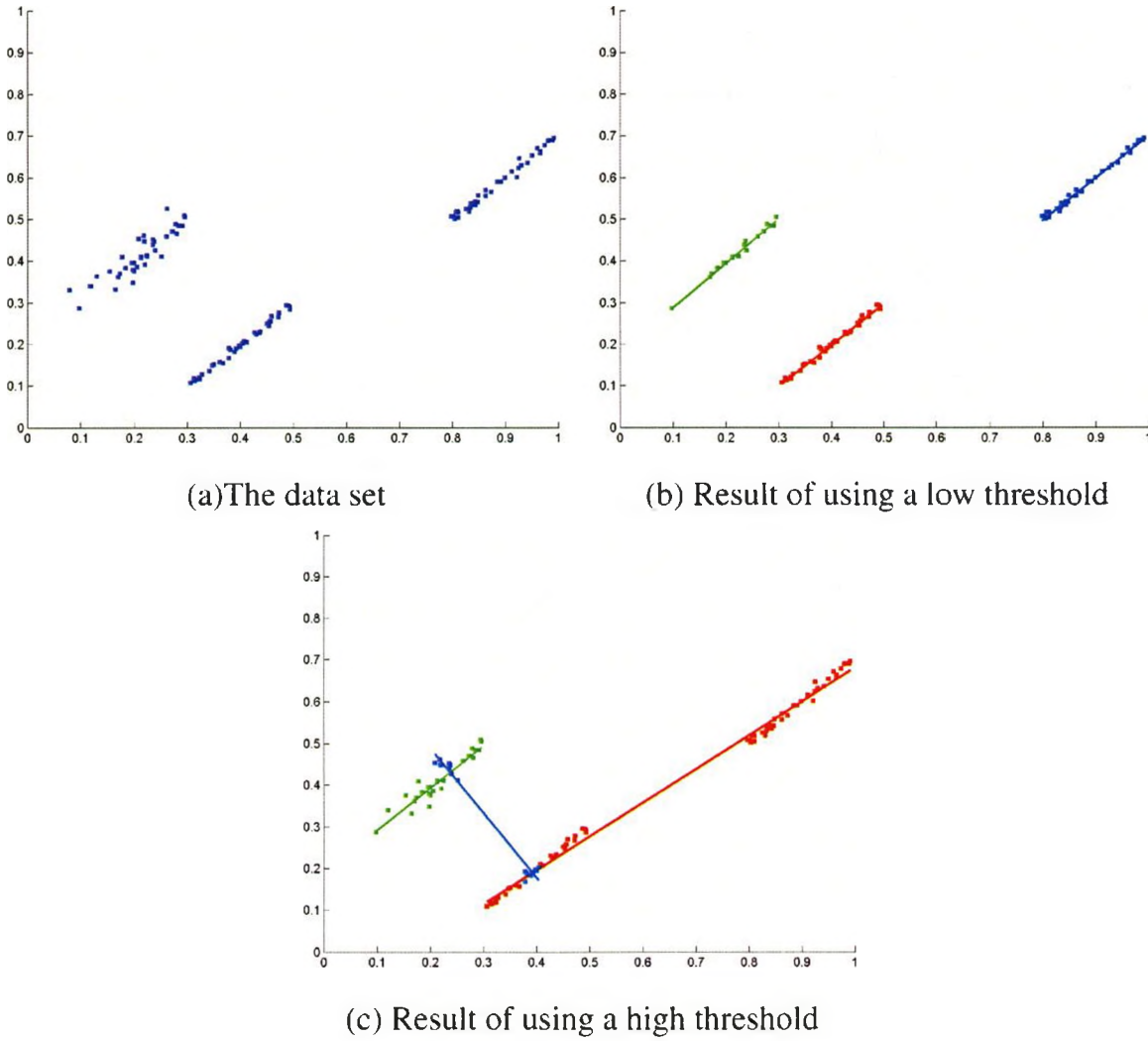


Figure 1.5: Figure (a) shows the data set contains 3 models, 40 inliers per model. Two models were perturbed with a low level of Gaussian noise and the other one with a higher level noise. MultiRANSAC results for using a low threshold and a high threshold are shown in Figures (b) and (c) respectively.

1.3.4.2 Ambiguous Points

Current RANSAC-based approaches can produce random results when assigning an ambiguous set of points to competing models (see Fig. 1.6). Figure 1.6(a) shows two models with spatially separated sets of inliers. Figure 1.6(b) shows the area in which a data point could fit for each. The inliers located inside the red circle (ambiguous set) fit both models. RANSAC-based approaches assign the ambiguous set of inliers to the first model to be

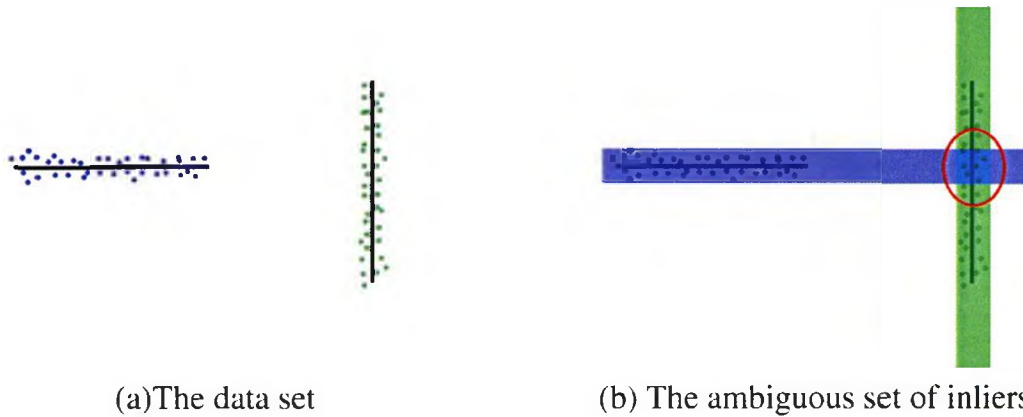


Figure 1.6: Competing models, Figure (a) shows two models with spatially separated sets of inliers. Figure (b) show for the horizontal line/model in which the blue area represents the set of all points that are at distance τ or less from the line/model. Any point that lies within that area fits the horizontal line. The same is true for the vertical line and the green area. The red circle depicts the ambiguous set.

detected.

Two models shown in Figure 1.7(a) are competing models. Figures (1.7)(b) and (c) show two different results obtained by using multiRANSAC to fit the given data points. Since the number of inliers is the same for both models, there is a 50% probability of obtaining either one of these results based on which model is detected first. The first model to be detected gets the ambiguous set. These two results show how RANSAC-based approaches depend upon the order in which the models are detected.

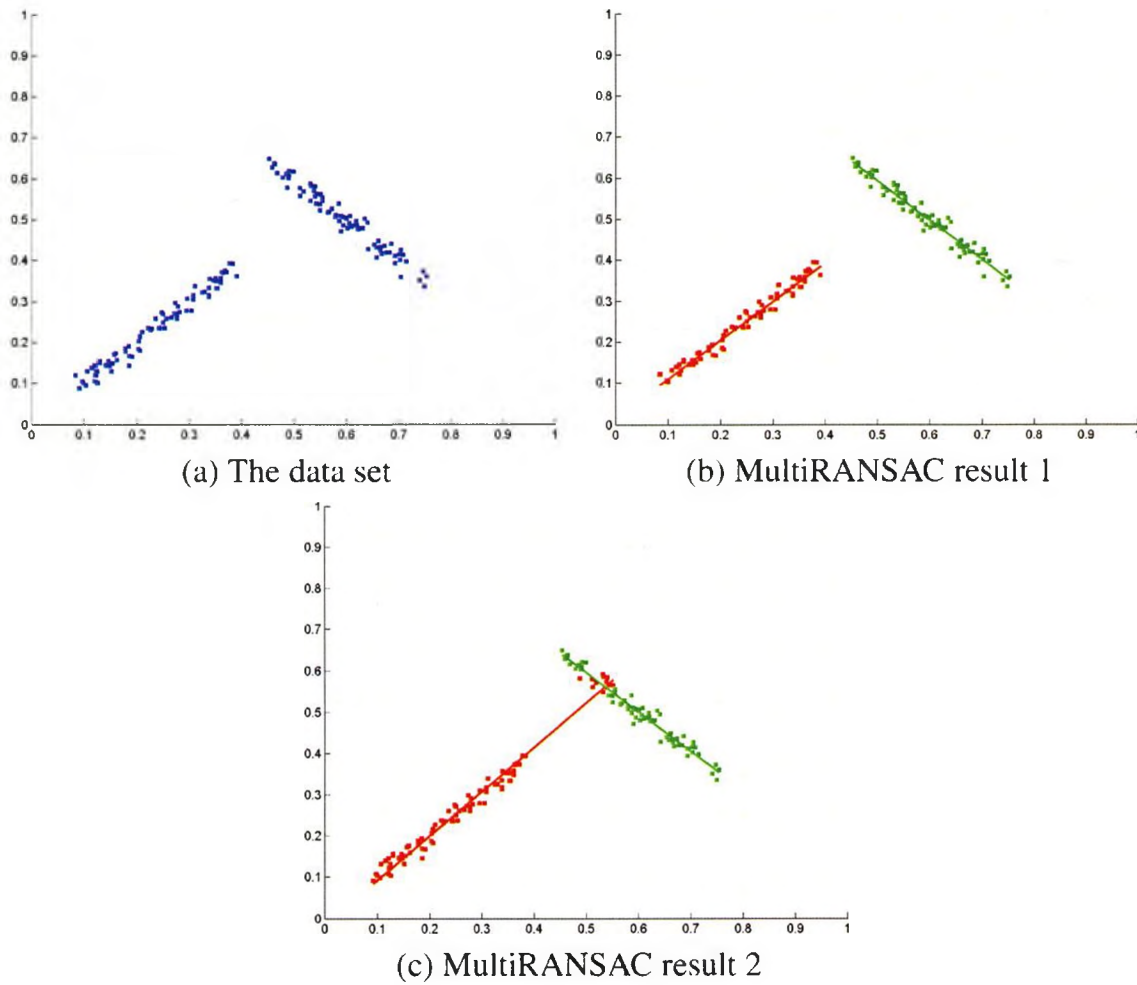


Figure 1.7: Figure (a) shows the data set contains two models, 40 inliers each. These two models are competing models. Both models were perturbed with the same Gaussian noise. Figure (b) shows the result of the ambiguous set being assigned to the correct model. Figure (c) shows the result of the ambiguous set being assigned to the wrong model.

1.4 From Thresholding to Regularization

SCO-RANSAC tries to find the smallest set of lines that fit all the data points. It does that by formulating the problem of classifying the inliers to the lines as a multi-labeling problem, in which each line is considered as a label and each data point is considered as a site, refer to section 1.4.1 for detailed explanation of the multi-labeling problem. Notice that unlike RANSAC-based methods, SCO-RANSAC neither rely on a threshold parameter τ in identifying the inliers nor it requires the prior knowledge of the number of models. Figures 1.8(a), (b), and (c) show the RANSAC inliers counting function that is maximized, and the RANSAC truncated geometric fitting error that is minimized as a final step, and the SCO-RANSAC geometric fitting error that is minimized, respectively.

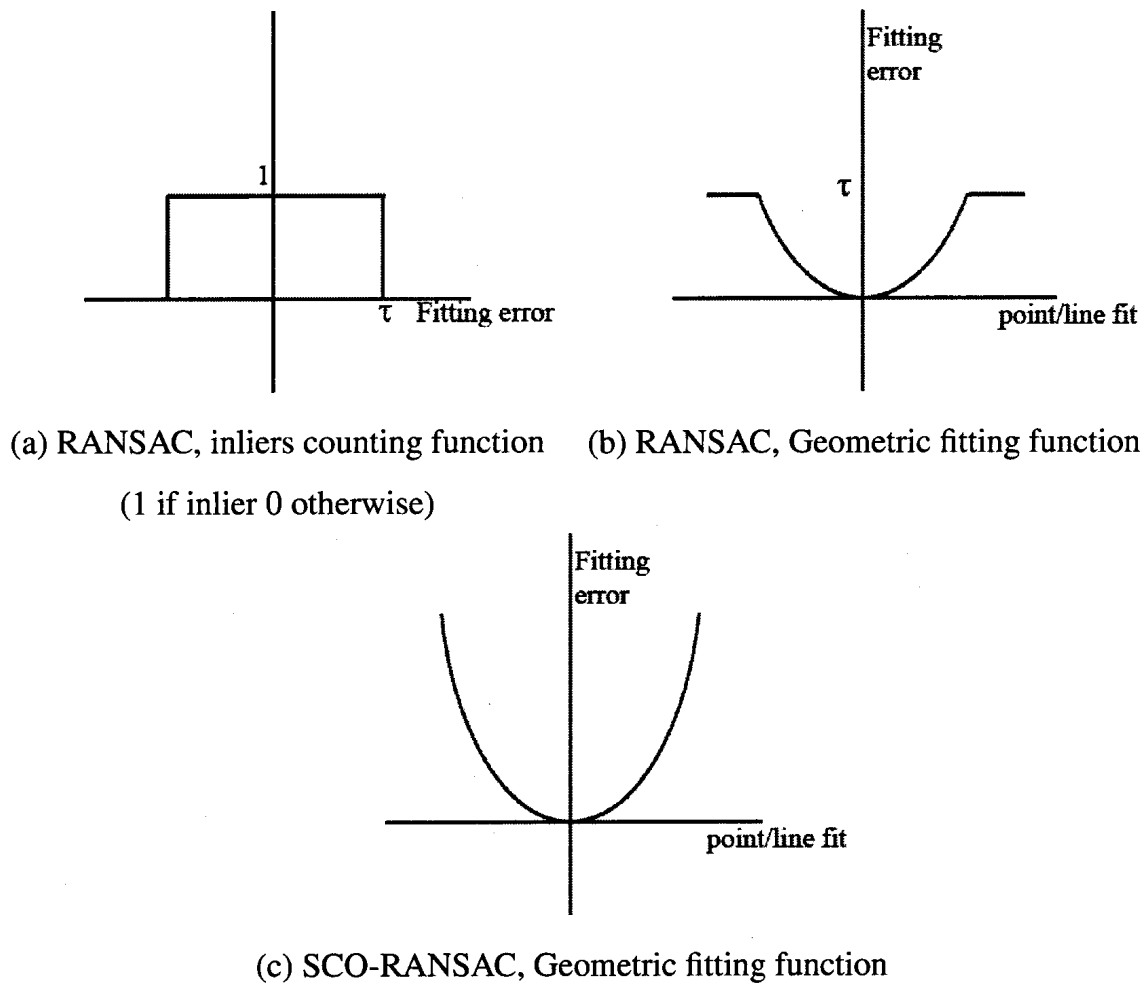


Figure 1.8:

1.4.1 Regularization and MRF-based models in Computer Vision

Many image analysis and computer vision problems can be posed as multi-labeling problems e.g., surface reconstruction from stereo [18], optical flow [22] and shapes from shading [23]. A Multi-labeling problem [31] is specified in terms of a set of sites P and a set of labels \mathcal{L} . For example, in the image restoration problem, the set of sites P are the image pixels and the set of labels \mathcal{L} are the gray scale values $0, 1, 2, \dots, 255$. The solution (a.k.a configuration) \mathbf{L} to the multi-labeling problem is a set of labels \mathcal{L} assigned to the sites P that minimizes the following energy function

$$E(\mathbf{L}) = \sum_{p \in P} ||L_p - p||$$

where L_p is the label assigned to the site p , and $||L - p||$ defines the penalty for assigning the site p to the label L . The previous energy term is referred to as the data term.

The aforementioned multi-labeling problems are ill-posed¹. The main idea for solving ill-posed problems is to restrict the class of admissible solution by introducing a suitable constraint (prior knowledge). This type of solution has been developed into a general framework, called regularization [31, 3], for a variety of low level vision problems. A generic constraint on this world is smoothness. Smoothness assumes that physical properties (in the neighborhood² of space \mathcal{N}) show some coherency, and generally do not change rapidly. For example, the surface of a table is flat. Since its early applications in vision [18, 22, 23] aimed to impose constraints, in addition to those from the data the smoothness prior has been one of the most popular prior assumptions. The smoothness prior is often expressed as the prior probability or equivalently an energy term measuring the extent to which the prior assumption is violated (discontinuity). The following equations show the

¹ A mathematical problem is ill-posed in the Hadamard sense if its (1) does not exist, (2) is not unique, or (3) does not depend continuously on the initial data.

²In image analysis, the most basic neighborhood systems are the 8-neighborhood and the 4-neighborhood systems, in which each pixel is connected to its 8 or 4 neighboring pixels, respectively.

energy function after adding the smoothness term

$$E(\mathbf{L}) = \sum_{p \in P} \|L_p - p\| + \lambda \sum_{\{p,q\} \in \mathcal{N}} V(L_p, L_q),$$

where $V(L_p, L_q)$ is the penalty of assigning the neighboring pixels p and q to the labels L_p and L_q , respectively.

Finding the optimal solution for the multi-labeling problem is *NP*-hard [7]. Geman and Geman [16] proposed an approach to regularization that is based on MRF models. In this approach the a priori knowledge is represented in terms of an appropriate probability distributions. In a MRF the value of a site depends only on the values within its given neighborhood. In this approach the best configuration maximizes some likelihood criterion such as the maximum a posteriori estimate MAP, for more details about the MAP-MRF models the reader is referred to [26, 16, 31].

1.4.2 Optimization and α -expansion

Since 1984 various methods have been applied to solve such MAP-MRF models in computer vision and other fields. Geman and Geman [16] proposed simulated annealing as a general method for obtaining an sub-optimal configuration for the MAP-MRFs. In the past few years more powerful optimization methods were introduced. Boykov, Veksler and Zabih [7] proposed the α -expansion graph cut algorithm that efficiently finds a local minimum within a factor of two from the optimal solution. The α -expansion allow large moves (namely expansion moves) which can simultaneously change the labels of arbitrarily large sets of pixels. In contrast to simulated annealing in which small moves (only one site changes its label at a time) were only allowed. Also, there are other new powerful methods such as loopy belief propagation [43] and tree-reweighed message passing [40]. A recent comparative study [35] between these new methods showed that the graph cuts [7] usually finds the best configuration in the least amount of time. For more details on how α -expansion finds the sub-optimal configuration for the multi-labeling problem the reader is referred to [7, 6].

1.5 Our Contribution

Our contribution is a new robust regression method for multi-model fitting. The main target of our work was to overcome the standard RANSAC-based methods' shortcomings: they are sensitive towards the threshold parameter; as pointed out in [36], they require a large number of initial labels; they do not guarantee a spatially coherent set of inliers for each model; and finally, they randomly assign one of the competing models to an ambiguous set of points.

Our algorithm attempts to remove these limitations. It is based on an insight that is, in practice, inliers for each model are often spatially coherent. All previous RANSAC-based methods ignore this. Like RANSAC-based methods, our method randomly samples data points to generate a set of hypotheses for the models (labels). Our method combines inlier classification and model fitting as a single energy minimization problem. Our objective function combines the geometric fit error and spatial coherence in one MRF-based energy. Moreover, it allows iterative re-estimation of models' parameters/inliers with a clear stopping criterion and convergence guarantees. We show that our new method, SCO-RANSAC, does not require a threshold parameter, empirically needs a smaller number of initial proposals compared to RANSAC-based methods, guarantees spatially coherent models, and often resolves ambiguity of points for competing models. In order to illustrate the generality of SCO-RANSAC, we demonstrate results for linear, affine and homographic models.

Chapter 2

SCO-RANSAC

Standard RANSAC-based methods depend on a threshold parameter, as described in section 1.2. Its value is important for detecting inliers for each model. Any misclassified point may significantly affect the accuracy of the estimated model. Accurate classification of points is even more critical when fitting multiple models, as the decision to assign a point to one of the models becomes more complex, especially when same points may fit more than one model (e.g., competing models).

We propose a new framework, Spatially Coherent RANSAC (SCO-RANSAC), for fitting multiple models. This framework combines model fitting and inlier identification into a single MRF-based regularization framework. In addition to considering some error function measuring geometric fit of the data points to a model as in RANSAC, our framework takes spatial coherence between the data points into account. Spatial relationships between sparse data points can be obtained via triangulation (e.g., Delaunay triangulation). Our regularization-based approach alleviates the dependence of the standard RANSAC-based methods on a single threshold parameter.

This chapter explains our multi-model fitting. In general, a model can be described by n real-valued parameters. For simplicity in this section, we illustrate the main idea in the context of synthetic multi-line fitting examples in 2D. In chapter 3, we validate the SCO-RANSAC approach by using real data for fitting other types of parametric models in higher dimensions.

2.1 Overview of the Main Idea

Similar to earlier RANSAC-based methods, SCO-RANSAC uses random sampling of minimal sample sets to create an initial finite set of model proposals (labels). Then an MRF-based regularization framework is used to assign labels (models) to data points. In this case, the energy combines geometric fit errors and spatial smoothness prior. As long as the number of labels is finite (≤ 10000 or so), such an approach can be handled by graph-based optimization methods. We use α -expansion [6] to identify the spatial support (consensus set) for each model. Once the consensus set of each model is known, the models (labels) can be re-estimated by minimizing only the geometric errors part of the energy. Clearly, one can continue in an **EM** style optimization of the energy until convergence by iterating over the two main steps: computing spatial support using α -expansion and re-estimating models parameters. In practice, given enough initial models, SCO-RANSAC converges on the average after three or six iterations, depending on whether the geometric error is linear or non-linear.

2.2 Initial Labels \mathcal{L}_0

For initial models (labels), SCO-RANSAC randomly samples minimal sample sets and generates a finite set of initial model proposals $\mathcal{L}_0 \subset \mathbb{R}^n$, where n is the number of parameters describing each model ($n = 2$ for lines, $n = 6$ for affine models, and $n = 8$ for homographic models). The idea of generating models by sampling the data points is borrowed directly from RANSAC [15]. Theoretically, the number of initial models $|\mathcal{L}_0|$ required by the RANSAC-based methods depends upon the number of data points, the number of outliers, the minimum number of points required to estimate a model, and the level of confidence that SCO-RANSAC will converge to a correct solution. In practice, we have found that SCO-RANSAC requires far fewer samples than the theoretical number: assuming that there exists W models in the data set with $|P|$ data points, and that the number of inliers of the models are $|P_1|, |P_2|, \dots, |P_W|$, such that $|P_1| \geq |P_2| \geq \dots \geq |P_W|$. In contrast to Sequential RANSAC that generates initial set of models for each model in

the data set, SCO-RANSAC generates only one set of initial models for all the models in the data set. Thus, in SCO-RANSAC, the number of initial models is chosen sufficiently high to ensure, with a probability of k , that at least one of the sampled minimal sample sets encloses data points that belong to the set P_W (the model with the smallest data support). Suppose that t is the probability that the selected data points belong to the set P_W . Then

$$t = \frac{\binom{|P_W|}{n}}{\binom{|P|}{n}} \quad (2.1)$$

and at least N iterations are required, such that

$$(1 - t)^N = (1 - k)$$

where

$$N = \frac{\log(1 - k)}{\log(1 - t)}. \quad (2.2)$$

It follows directly from (2.1) and (2.2) that the number of initial proposals $|\mathcal{L}_0|$ required by SCO-RANSAC depends upon the size of the smallest data support $|P_W|$ and the total number of data points in data set $|P|$.

2.2.1 Outlier Labels

Our approach is based on assigning labels to all data points. In the previous section we described how a set of possible labels (models) is sampled. Our problem is that many outliers will not fit well into any of the sampled models¹ [see Fig. 2.1]. Yet, such outliers will still have to be assigned to some model (label). Clearly this can corrupt the set of inliers for such a model (label). In order to avoid this situation, we can introduce a label (model) for each data point, we call these labels outlier labels.

Definition 3 *An Outlier label of a data point p is a unique label for p that fits it with zero error and fits all the other data points with ∞ error. The consensus set of p outlier label is either empty (p is an inlier of another model) or contains only p (p is an outlier).*

¹unless the number of samples is nearly exhaustive

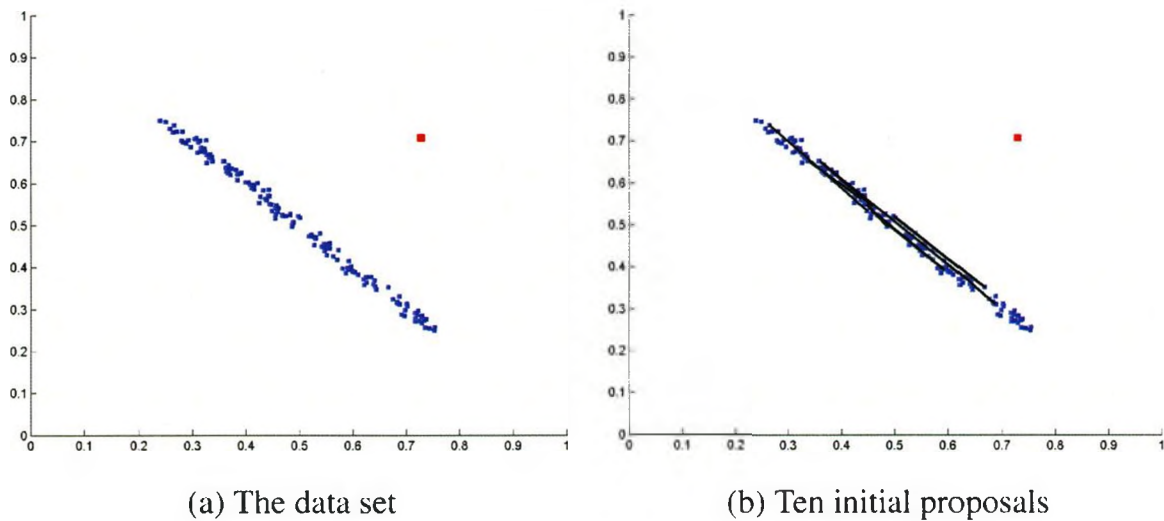


Figure 2.1: Figure (a) shows a data set with 150 inliers and one outlier (shown in red). Figure (b) shows ten initial proposals. As you can see its unlikely for the outlier to get sampled unless the number of initial proposals was really large. To avoid this we introduce an outlier label for each data point.

By introducing an outlier label for each data point to the set of initial proposals, we guarantee that the SCO-RANSAC will always find a model that fits any data point with zero error, whether it was sampled or not. Although the solution that assigns each data point to its outlier model will have a zero sum of fitting errors, that solution will also maximize the number of spatial discontinues. This is not a possible solution in our case, since we are minimizing both the geometric fit error and the number of spatial discontinues. Thus, it is safe to introduce the synthetic models.

In graph cuts outliers are commonly dealt with by using a robust measure for the data term. For example its possible to replace the quadratic error by a truncated linear or truncated quadratic error measure. But using a truncated linear or truncated quadratic measure adds an extra parameter, the truncation threshold. As you can see, using outlier labels does not introduce another parameter.

2.3 Energy Formulation

Once the initial finite set of proposed models $\mathcal{L}_0 \subset \mathbb{R}^n$ is known [see Fig. 2.2(a)], we can estimate their spatial support. We use an MRF-based regularization framework and α -expansion optimization [7] to assign models to data points. The set of current models in \mathcal{L}_0 is interpreted as the set of current labels. Assume that P is a set of data points and that $L_p \subset \mathbb{R}^n$ is a label (model) assigned to a given data point $p \in P$, then the SCO-RANSAC method estimates models and their spatial support (inliers) by optimizing the following energy of labeling $\mathbf{L} = \{L_p | p \in P\}$:

$$E(\mathbf{L}) = \sum_p \|p - L_p\| + \lambda \cdot \sum_{(p,q) \in \mathcal{N}} w_{pq} \cdot \delta(L_p \neq L_q). \quad (2.3)$$

The first term $\|p - L\|$ in (2.3) describes the geometric error between point p and model L . For example, line fitting examples in this section measure deviation between 2D point $p = (x, y)$ and line $L = (a, b)$ based on “vertical shift”:

$$\|p - L\| = (y - ax - b)^2.$$

There are other ways to measure geometric errors $\|p - L\|$ as well. For example, one can measure distance from p to the nearest point that perfectly fits model L . Robust (truncated) measures are also possible. The data term $\|p - L\|$ corresponds to the log-likelihood $\ln \Pr(p|L)$ when energy (2.3) is interpreted as an MRF-based posterior energy. Thus, the use of quadratic distance for $\|p - L\|$ is equivalent to assuming Gaussian distribution for errors. Clearly, optimal labeling L for (2.3) depends on the specific choice of geometric measure $\|p - L\|$.

The second term of energy (2.3) is a smoothness prior. It assumes some specific neighborhood system \mathcal{N} for the data points. For example, the neighborhood system can be based on a triangulation of points (see Fig. 2.3). In this work, we use the Potts model [7], where $\delta(\cdot)$ is 1 if the specified condition inside the parentheses holds, and 0 otherwise. Weights w_{pq} set discontinuity penalties for each pair of “neighboring” data points. For example, the synthetic line fitting examples in this chapter used weights w_{pq} inversely proportional to

the distance between points p and q , because closer points are more likely to fit the same model. Besides the Potts (piece-wise constant) prior, one can also consider piece-wise smooth priors. Such priors will allow small variations in model parameters between data points.

2.4 Inlier Classification and Model Re-estimation

Energy (2.3) can be minimized using an α -expansion algorithm [7] for labels $a \in \mathcal{L}_0$. In this case, it is possible to interpret α -expansions as a competition among models/labels for spatial support; models that best fit the data points find the largest number of spatially coherent “inliers”, while most of the “erroneous” models get no inliers.

Once inliers are computed, models/labels in $\mathcal{L}_0 \subset \mathbb{R}^n$ with non-empty sets of inliers can be re-estimated as follows. Note that the first term of energy (2.3) can be represented as

$$\sum_p \|p - L_p\| = \sum_{L \in \mathcal{L}_0} \sum_{p \in P_L} \|p - L\|$$

where $P_L = \{p \in P | L_p = L\}$ denotes a set of inliers for label L . Clearly, we can minimize this expression by re-estimating parameters of each model $L \in \mathcal{L}_0$:

$$\hat{L} = \arg \min_l \sum_{p \in P_L} \|p - l\|. \quad (2.4)$$

We replace each label L with a non-empty support P_L by label $\hat{L} \in \mathbb{R}^n$, which has a better fit to points in P_L . Finally, after discarding all labels with no inliers, we obtain a new set of labels, \mathcal{L}_1 . Note that this operation does not affect the second (smoothness) term in (2.3) unless two labels, L and L' , become equal after re-estimation $\hat{L} = \hat{L}'$ (in this case, the smoothness energy also decreases). Clearly, the described operation of changing the set of labels

$$\mathcal{L}_0 \rightarrow \mathcal{L}_1$$

can only decrease the energy (2.3).

There are many known methods for optimizing the sum of geometric errors $\|p - L\|$ in

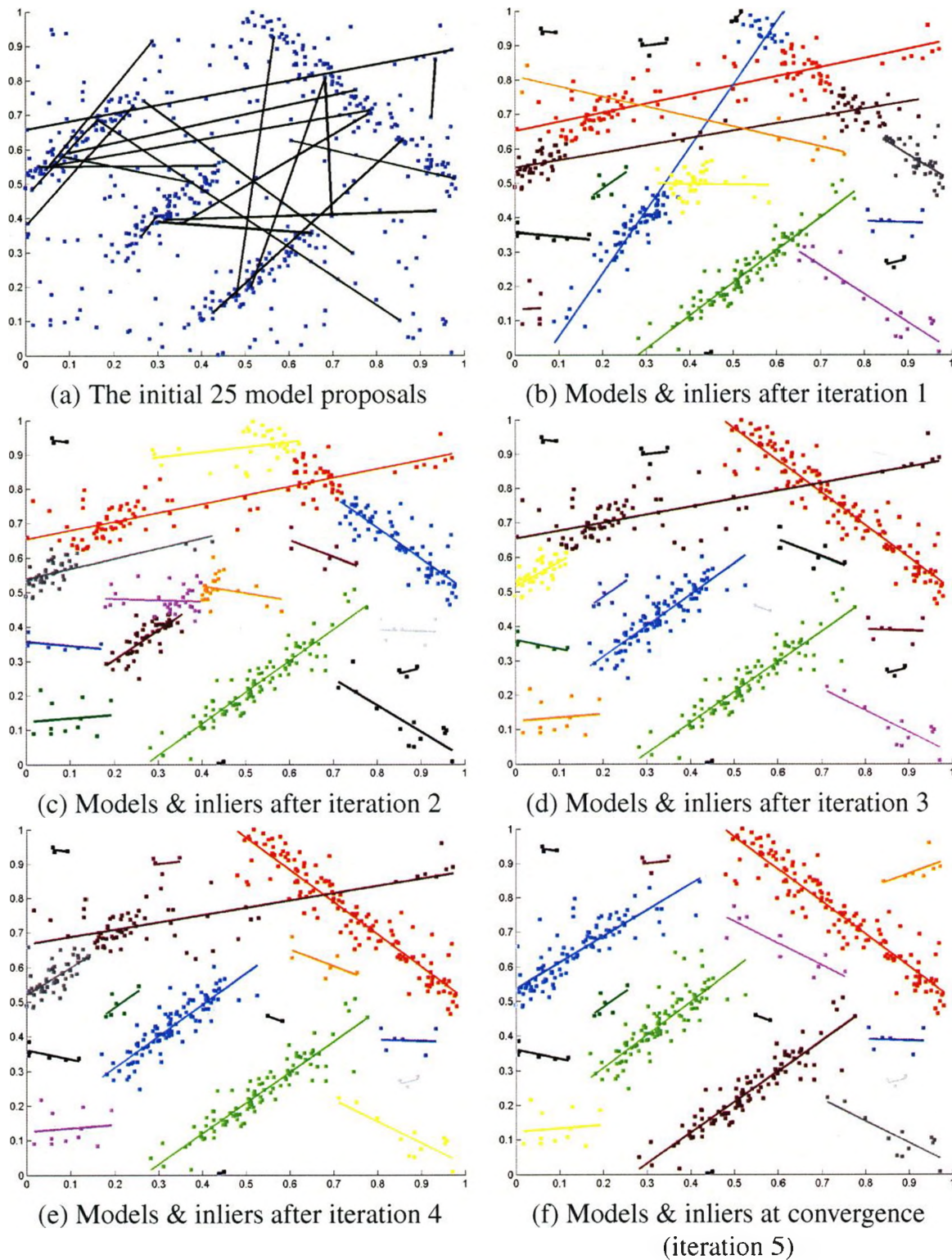
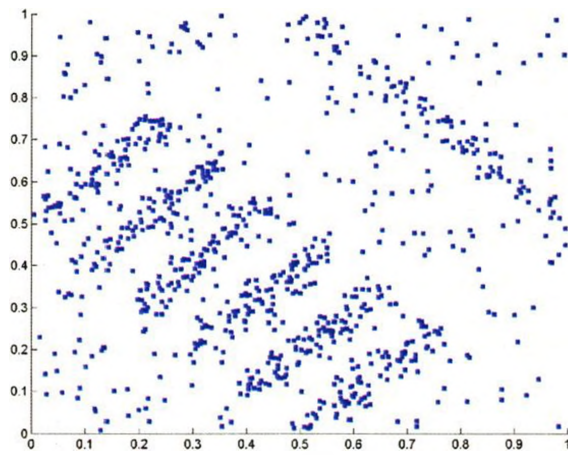
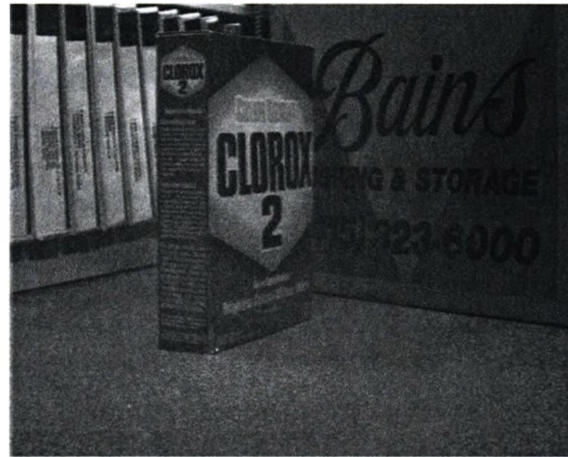


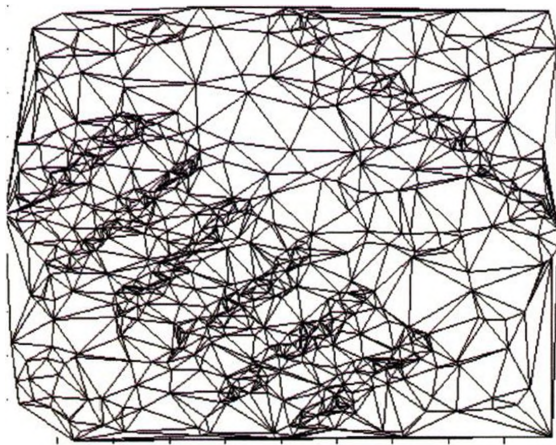
Figure 2.2: Illustrations of SCO-RANSAC's iterations. Figure (a) show the proposals generated by random sampling. Figures (b) through (f) show the re-estimation of models and their inliers. This is an illustrative example. SCO-RANSAC only produces a reasonable result with a certain probability. This probability increases as more initial proposals are used.



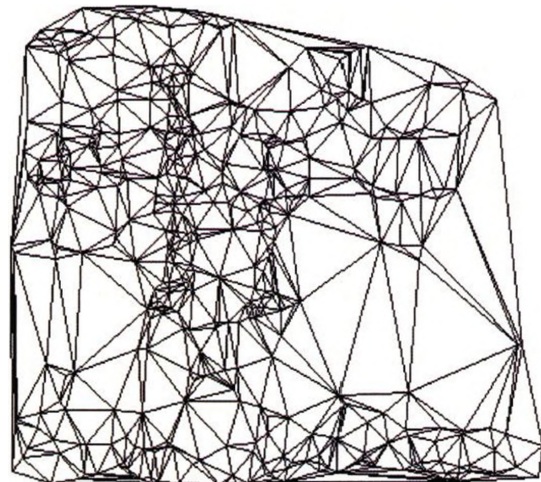
(a) Lines data set



(b) Clorox data set from Birchfield and Tomasi [4]



(c) Triangulation of the data set points



(d) Triangulation of points of interest used in estimating the models

Figure 2.3: Delaunay triangulations of data points. Other techniques can be used, particularly for higher dimensional data.

(2.4). The optimization method may depend upon the specific choice of measure $\|p - L\|$. For example, to minimize the sum of squares of vertical-shift errors in our line fitting examples, we use standard SVD-based methods to solve for the least squares fitting. One could also minimize the sum of squares of the perpendicular offsets. A large number of other examples of geometric and algebraic error measures $\|p - L\|$, and different methods for optimizing them, are widely discussed in computer vision literature. Our approach can incorporate many of these error functions $\|p - L\|$. Figure 2.2 (b) illustrates clusters of inliers and re-estimated models \mathcal{L}_1 obtained in the two separate steps described above: α -expansion (inlier classification) and geometric error minimization (model re-estimation).

2.5 Our Algorithm

Both “inlier classification” and “model re-estimation” steps described in the previous section decrease energy (2.3). Thus, we can iterate over these steps until convergence [see Fig. 2.2 (b)–(f)]. We can stop the iterations when a new round of α -expansion does not change inliers. As soon as the spatial support of the current models (labels) stops changing, re-estimation of the models (2.4) cannot improve the geometric error term. The SCO-RANSAC algorithm is summarized in Algorithm 5.

Algorithm 5 SCO-RANSAC

input:

- P – The data set of points
- n – Minimum number of points required to estimate the model parameters
- k – The probability of SCO-RANSAC success in finding all the models
- t – The probability of randomly sampling two inliers of the smallest model

output :

- \mathcal{L} – The models found
-

- 1: $N = \frac{\log(1-k)}{\log(1-t^n)}$
 - 2: $\mathcal{L} \leftarrow$ Randomly sample N MSSs {**Initialization**}
 - 3: **repeat**
 - 4: $\mathcal{L} \leftarrow$ Run α -expansion for energy (2.3) {**Inlier classification**}
 - 5: $\mathcal{L} \leftarrow$ Solve (2.4) to obtain new model parameters {**Model re-estimation**}
 - 6: **until** the energy does not decrease
-

2.6 General Properties

2.6.1 Noise and Outliers

In this section, we will compare SCO-RANSAC to multiRANSAC and Sequential RANSAC using synthetic examples based on the seven models/lines shown in Figure 2.4. Six of these lines are parallel to each other (shown in black) and the last model (shown in red) is a competing model. Most of the inliers of the competing model also fit into one of the parallel models. Each of the six parallel models will contribute 80 points and the competing model will contribute 90 points. The synthetic examples were generated by adding different levels of Gaussian noise to the points and a different number of uniformly distributed outliers were added to the data set.

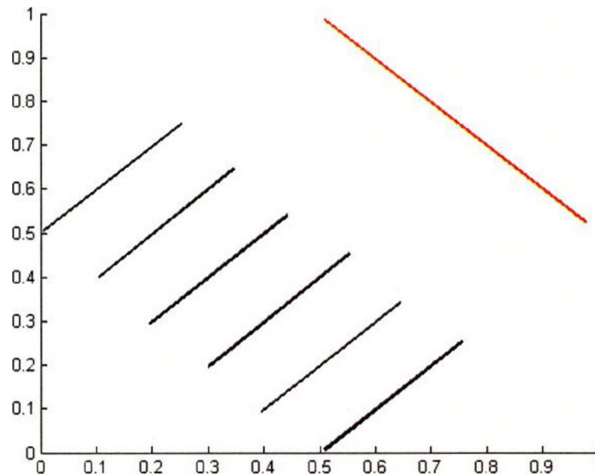


Figure 2.4: The seven Original models used in generating the synthetic data sets.

In the following experiments, we will show all the models found by Sequential RANSAC and SCO-RANSAC without any post processing that would eliminate the weak models. The number of models to be found is a parameter to the multiRANSAC method, thus it will always find only seven models.

We tuned the multiRANSAC and Sequential RANSAC parameters for each data set in order to get the best possible results for these methods. For SCO-RANSAC, the normalization factor λ and the number of initial proposals were the same in all of the following experiments. This shows that SCO-RANSAC parameters were more tolerant to noise level

variation than the parameters of the RANSAC-based methods that required retuning for each data set.

Low Noise Level

As can be seen in Figure 2.5(b)–(c) the results obtained from the RANSAC-based methods were fine for the parallel models; however the competing model lost some of its ambiguous points to the parallel models. Figure 2.5(d) shows that SCO-RANSAC was able to identify the data points for all the models correctly. Notice how the SCO-RANSAC assigned the added outliers to models as if they were actually inliers. These models could easily be filtered out based on the variance of a model's error fit or the number of points assigned to it.

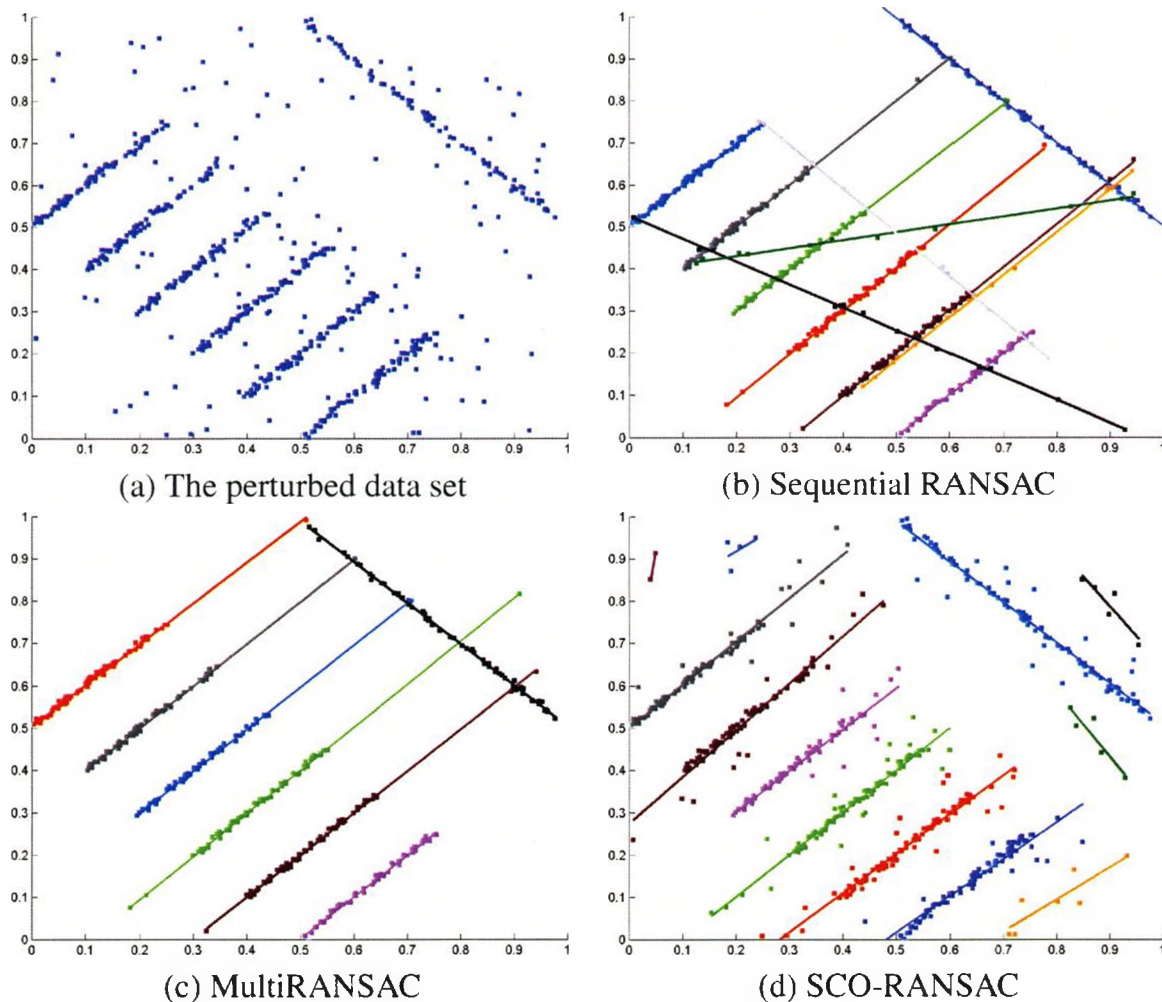


Figure 2.5: Comparing the results for fitting lines to noisy data points. The data points were perturbed with a low level of Gaussian noise ($\sigma = 0.005$) and 120 outliers were added.

Medium Noise Level

For the data set shown in Figure 2.6(a), Sequential RANSAC was still able to identify the models, but the competing model lost more of its inliers to the other models. MultiRANSAC failed to identify the parallel models, as it uses a greedy algorithm for selecting the models that cover the maximum number of inliers, which is not always correct. In this experiment, the number of inliers covered by a model across the parallel models is more than those covered by a parallel model; thus, multiRANSAC fails. SCO-RANSAC was able to correctly identify the data support for all the models.

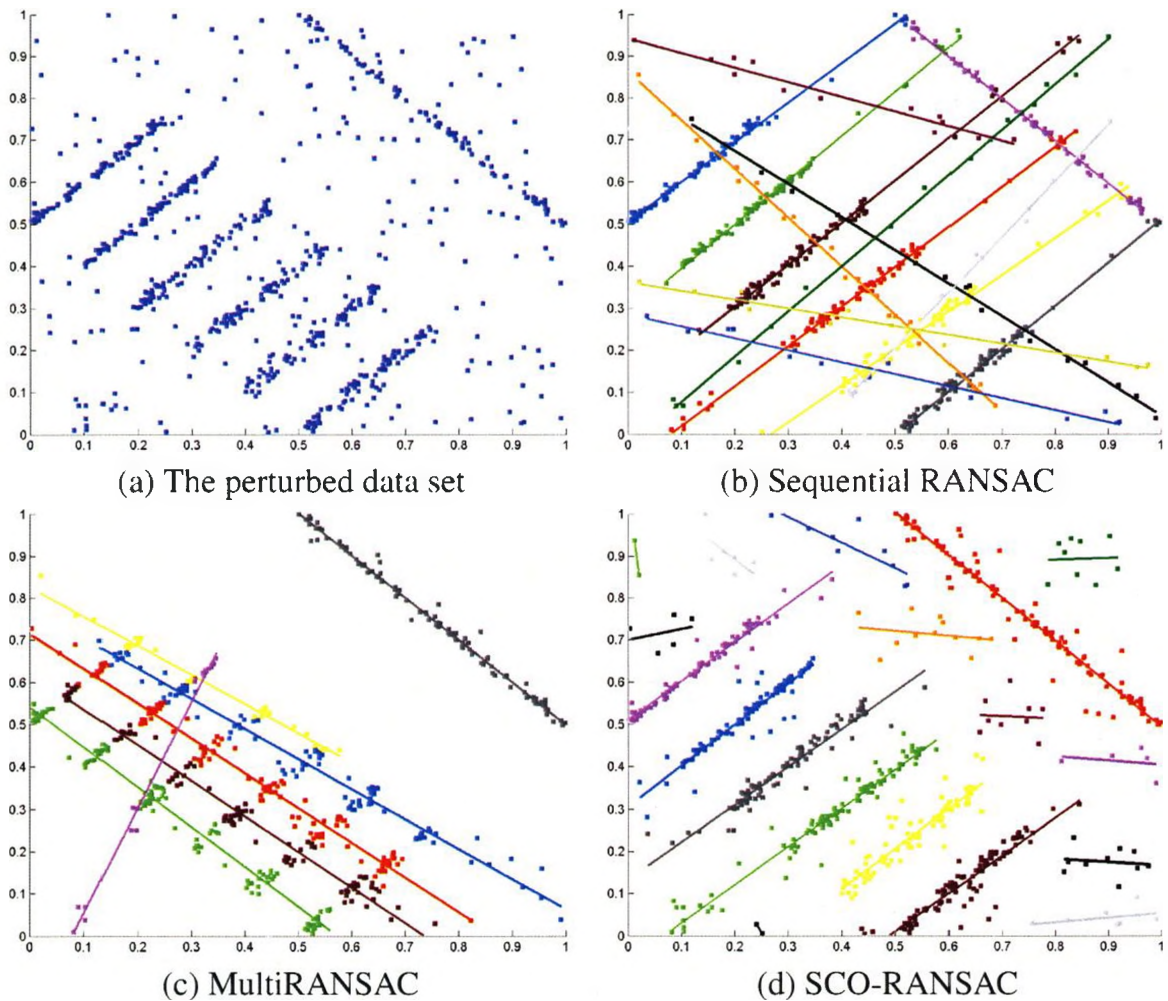


Figure 2.6: A Comparison of the results for fitting lines to noisy data points. The data points were perturbed with a medium level of Gaussian noise ($\sigma = 0.01$) and 200 outliers were added.

High Noise Level

The data set shown in Figure 2.7(a) is the result of perturbing the data points with a high level of Gaussian noise and adding 300 outliers. Both Sequential RANSAC and multi-RANSAC failed to identify the parallel models. In this experiment, the perturbed data points of the parallel models started merging into one another, forming almost one cluster of points. SCO-RANSAC was able to identify all the models, as it makes use of the distance between two neighboring points. The discontinuity penalties w_{pq} in (2.3) were set to be inversely proportional to the distance between each pair of neighboring data points. Thus, SCO-RANSAC was able to efficiently discriminate between the data points of each model.

Although the competing model was identified by all the methods, only SCO-RANSAC was able to correctly identify the data points for that model. Both RANSAC-based methods assigned some of the ambiguous data points to the wrong model (one of the parallel models).

As mentioned in [45], the number of iterations required by the multiRANSAC algorithm becomes large when the standard deviation of the Gaussian noise that perturbs the data points becomes larger. To cope with this problem, we limited the number of iterations to 1,000. In each iteration, multiRANSAC randomly sampled seven models; thus, in total, multiRANSAC randomly sampled 7,000 models in this experiment. Sequential RANSAC found 15 models, and for each model it randomly sampled 700 models. Thus, in total, Sequential RANSAC randomly sampled 10,500 models. The number of initial proposals used by SCO-RANSAC in this experiment was 500.

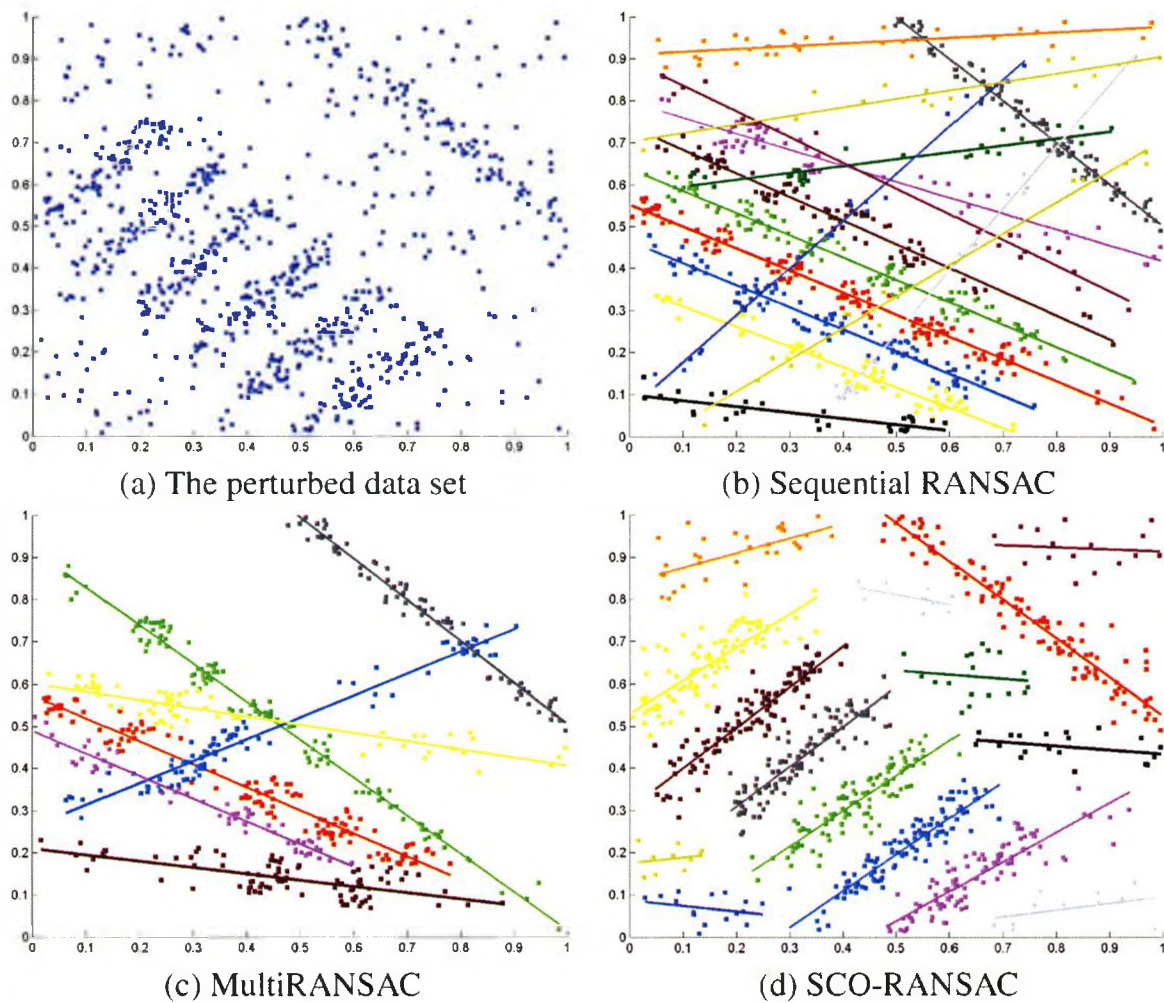


Figure 2.7: A comparison of the results for fitting lines to noisy data points. The data points were perturbed with a high level of Gaussian noise ($\sigma = 0.02$) and 300 outliers were added.

2.6.2 Number of Initial Samples

The data set shown in Figure 2.8(a) was obtained by perturbing the data points with a very high level of Gaussian noise ($\sigma = 0.025$) and 500 outliers were uniformly distributed over the domain. In this experiment, outliers constitute 87% of the data set. Figures 2.8(b) and (c) show SCO-RANSAC result for 500 and 6,000 initial proposals respectively. These results show that SCO-RANSAC is robust to noise and outliers once it is provided with a sufficient number of initial proposals.

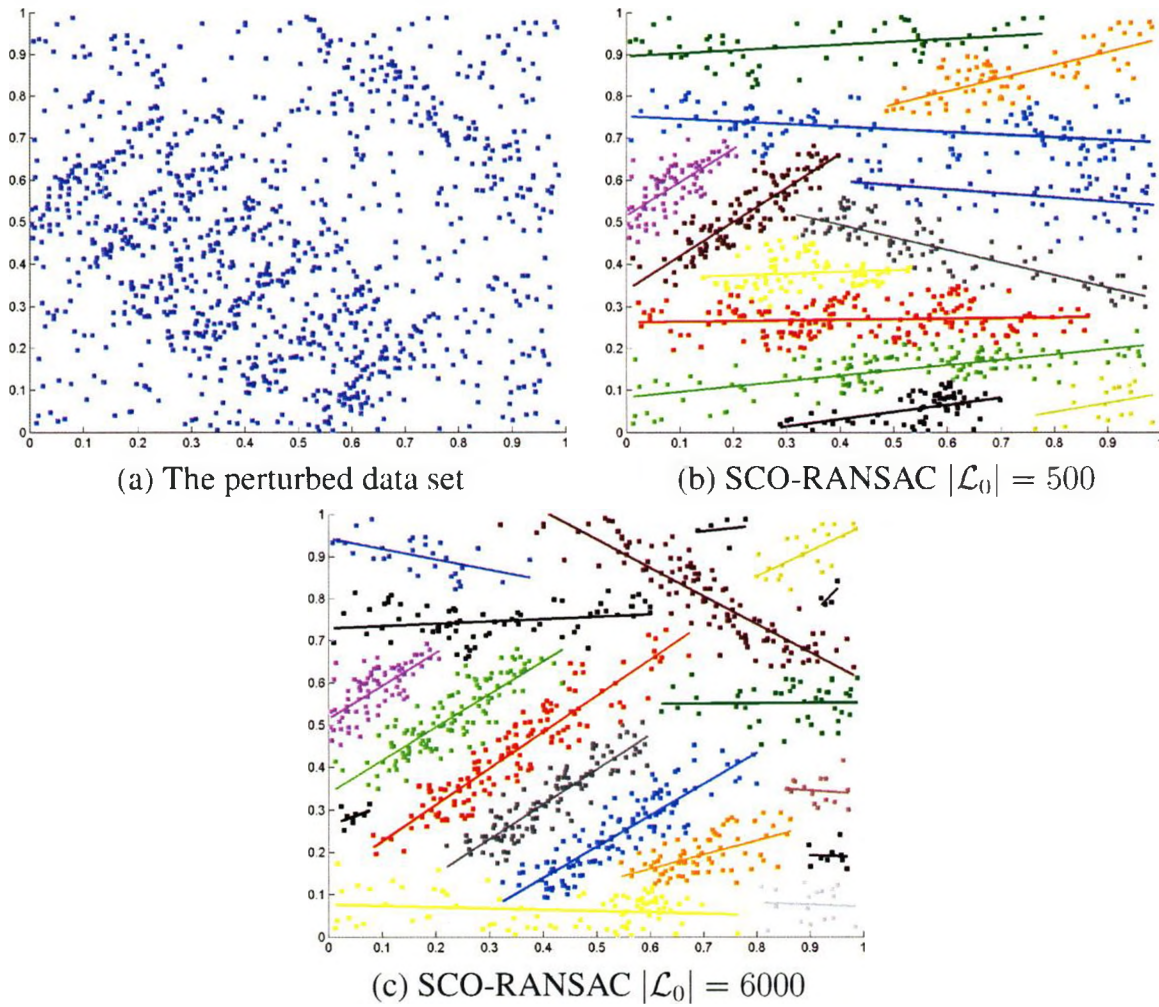


Figure 2.8: The results for using SCO-RANSAC for fitting lines to noisy data points. The data points were perturbed with a very high level of Gaussian noise ($\sigma = 0.025$) and 500 outliers were added.

2.6.3 Accuracy of Estimated Models

In this section, we will compare the accuracy of the model parameters estimated by SCO-RANSAC and RANSAC to the ground truth model. SCO-RANSAC could be used to fit a single model, simply after SCO-RANSAC converges select the strongest model e.g. the model with the largest number of inliers. We will only show the accuracy plot for only one of the line parameters. In the case of synthetic lines, the two model parameters are the slope m , and the intercept b . Figure 2.9 shows the average absolute error of slope (as y -axis) over 400 different data sets, for both SCO-RANSAC and RANSAC, using different number of randomly sampled models (as x -axis). In general SCO-RANSAC estimate more accurate model parameters compared to those estimated by RANSAC. What is interesting is that SCO-RANSAC gives highly accurate model parameters for small number randomly sampled models.

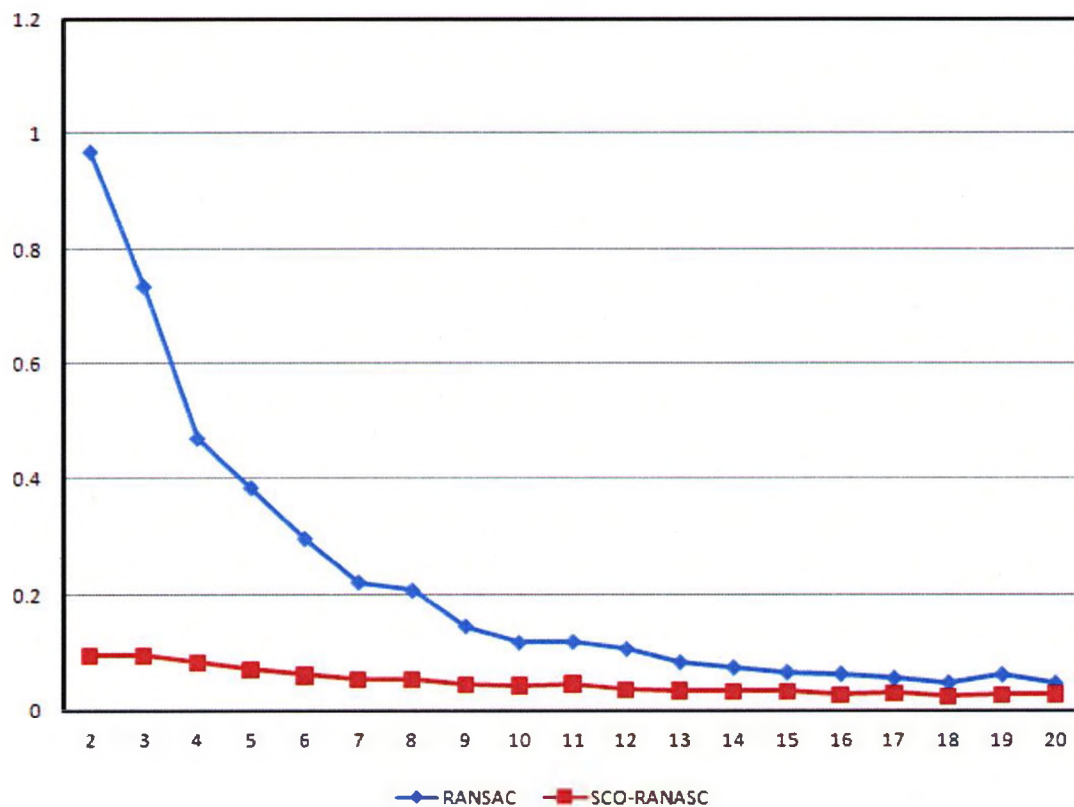


Figure 2.9: Error vs. the number of randomly sampled models plot

2.6.4 Performance

Let n and W denote the number of data points and the number of models in P , respectively. For each model, the Sequential RANSAC generates N random samples and computes their consensus sets. Thus, the Sequential RANSAC [39] algorithm's complexity is $O(nNW)$, when the number of models in the data is known.

As described in [45], multiRANSAC requires \hat{N} iterations. In each iteration, it randomly samples W models and computes their consensus sets. Thus, multiRANSAC's complexity is $O(n\hat{N}W)$. In [45], the author's have shown that multiRANSAC is more complex than Sequential RANSAC ($\hat{N} > N$), especially when the noise level increases.

SCO-RANSAC's complexity depends upon the number of initial proposals and the number of edges and vertices in the neighborhood graph. Assuming that the number of initial models $|\mathcal{L}_0|$ is \tilde{N} , the first step in an SCO-RANSAC iteration is to apply a -expansion on each label. The key step in a -expansion is to solve the min-cut/max-flow problem. The theoretical worst case complexity for using the push-relabel algorithm [17, 5] to find the min-cut in an m -edge, n -vertex graph is $O(n^2\sqrt{m})$. Thus, the complexity of SCO-RANSAC's first step becomes $O(\tilde{N}n^2\sqrt{N_e})$, where N_e is the number of edges.

Lemma 1 *Given a set P of n points, any triangulation of P has the same number of edges, ($N_e = 3(n - 1) - N_h$), where N_h is the number of points on the convex hull of P [11, 25].*

Thus, the complexity of the first step can be simplified to $O(\tilde{N}n^2\sqrt{n})$. The complexity of the second step, estimating the models' parameters, is less complex than the first step. As shown in Table (2.1) on average, SCO-RANSAC empirically requires 3 iterations to converge. Thus, SCO-RANSAC complexity is $O(\tilde{N}n^2\sqrt{n})$. Compared to the other methods, SCO-RANSAC's complexity is not far off, as it does not depend on the number of models in the data. The empirical results also show that \tilde{N} is much smaller than N and \hat{N} . In the example shown in Figure 2.7 \tilde{N} is by smaller than N and \hat{N} by an order of magnitude. Later on in section 3.2.3.2 we will present an example for estimating more complex models in which \tilde{N} is smaller than \hat{N} by two orders of magnitude.

Run	Iteration				
	0	1	2	3	4
1	500	17	15	15	15
2	500	16	15	14	14
3	500	16	15	13	13
4	500	16	16	15	15

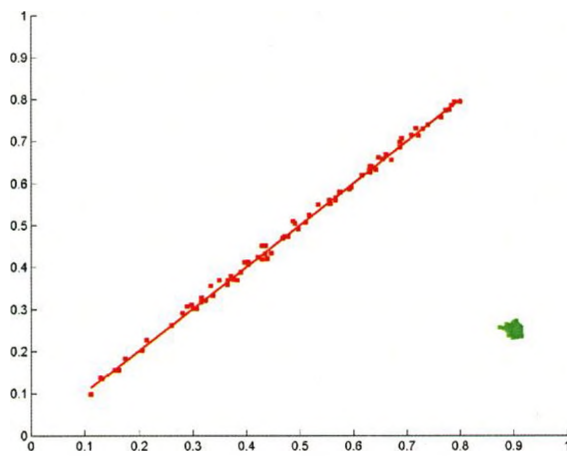
Table 2.1: Convergence, this table shows the number of labels \tilde{N} remaining after each iteration for four independent runs of SCO-RANSAC using the data set shown in Figure 2.7(a). Notice how the first iteration eliminated most of the hypotheses.

2.6.5 Overcoming the Limitations of Previous Methods

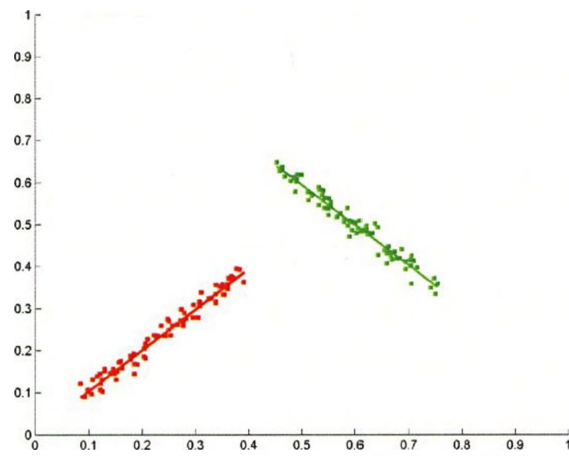
Figure 2.10(a) shows SCO-RANSAC result for the data set shown in Figure 1.2(a). Basic RANSAC failed to identify the true model due to the clustered outliers, but SCO-RANSAC was able to find both models. Figure 2.10(b) shows SCO-RANSAC result for the data set shown in Figure 1.7. SCO-RANSAC assigns the ambiguous set of inliers in a deterministic way. Figure 2.10(c) shows that SCO-RANSAC was able to correctly identify all the inliers for the case in which the models were perturbed by different noise levels (see Fig. 1.5 for multiRANSAC result for the same data set).

In summary, SCO-RANSAC was able to identify the models and classify the data points among them without any problems, as it takes spatial coherence into account and makes use of the distance between neighboring points. As the amount of noise increases, multiRANSAC [45] becomes very sensitive to the threshold parameter and estimates incorrect models. Also, as mentioned in [45], as the noise level increases, multiRANSAC requires that the number of iterations to be increased. Sequential RANSAC [39] was no better than multiRANSAC for high noise levels. SCO-RANSAC demonstrated robustness to the amount of noise. It also ran in about the same amount of time in the experiments provided. Moreover, SCO-RANSAC required much fewer random samples to converge to the correct models. Based on the results presented in this section, we conclude that SCO-RANSAC is

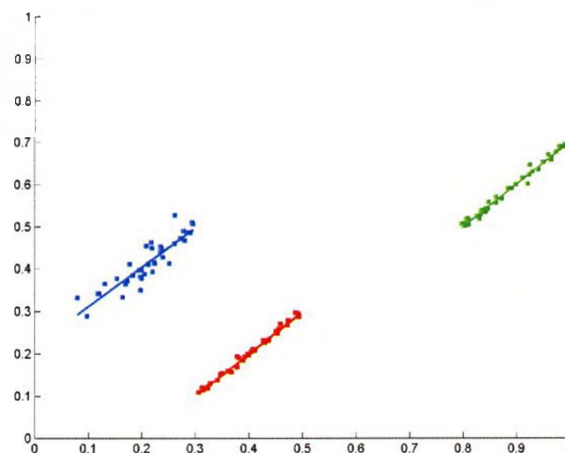
robust to noise and outliers, as it accounts for spatial coherence between data points.



(a) SCO-RANSAC result in the case
of clustered outliers



(b) SCO-RANSAC result in the case
of competing models



(c) SCO-RANSAC result in the case of
models with different noise levels

Figure 2.10: This figure shows SCO-RANSAC results for the data sets presented in Figure 1.2(a), Figure 1.7(a) and Figure 1.5(a), respectively. These results demonstrate that SCO-RANSAC was able to overcome the pitfalls of the RANSAC-based approaches, and the clustered noise situation in which basic RANSAC failed.

2.7 Merging Spatially Overlapping Models

The case of overlapping models was referred to in the introduction section 1.1, as a case in which our main assumption is violated. The example in Figure 2.11 contains two overlapping models, the consensus sets of these two models overlap with one another. Obviously, at least one of the models cannot get spatially connected support region. As shown in Figures 2.11 (b) and (c) SCO-RANSAC is not guaranteed to merge the two spatially disconnected models, even if their parameters are close. In the case of spatially disjoint data support, one can use a simple post processing step that can merge any two models if the resulting model increases the sum of geometric fit errors by no more than a predefined threshold [see Fig. 2.11(d)]. This operation can be justified by adding an extra term to energy (2.3), penalizing the number of labels $|L|$:

$$E(\mathbf{L}) = \sum_p ||p - L_p|| + \lambda \cdot \sum_{(p,q) \in \mathcal{N}} w_{pq} \cdot \delta(L_p \neq L_q) + \gamma \cdot |L|.$$

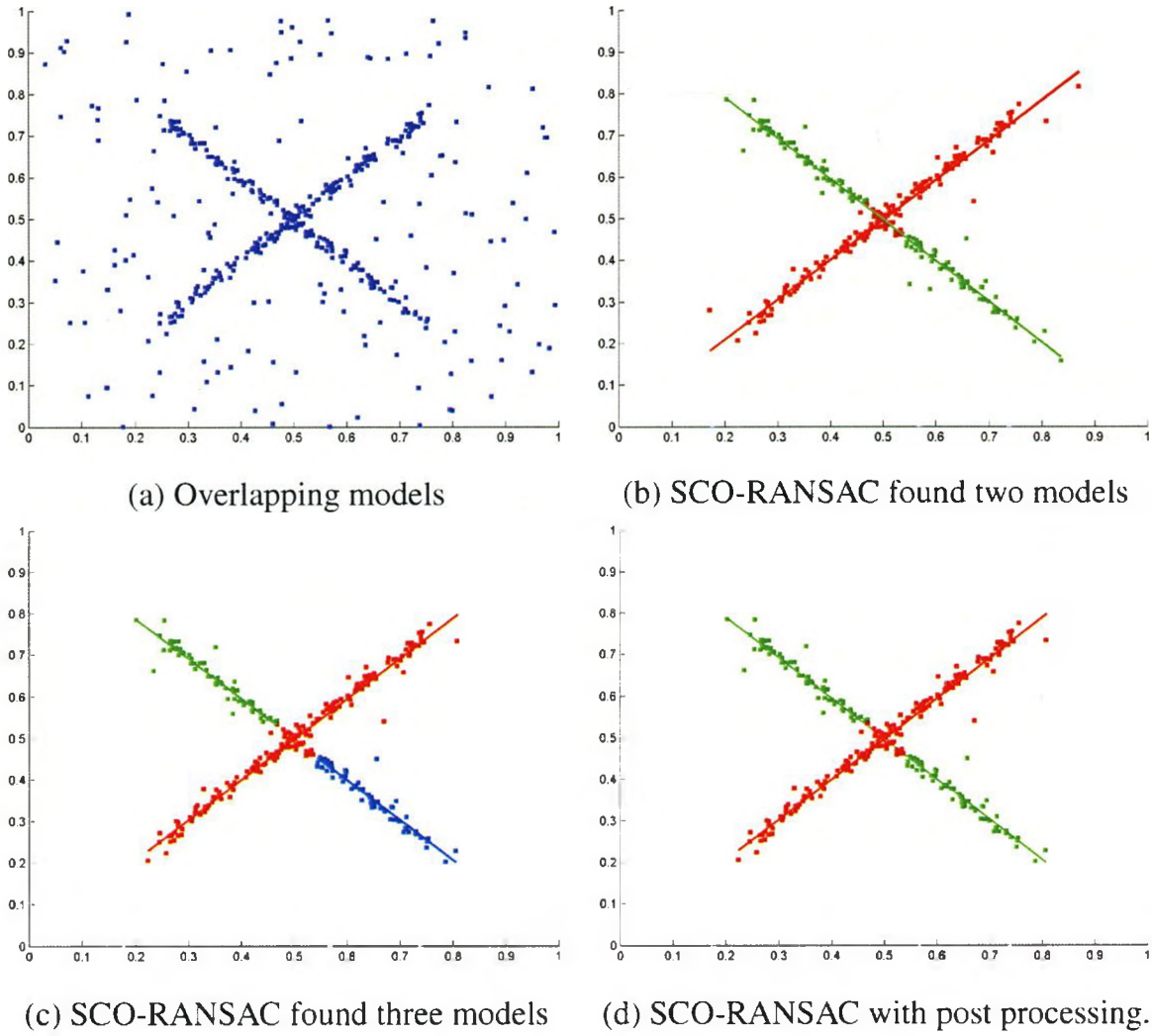


Figure 2.11: This figure illustrates the case of spatially overlapping models. Figure (a) shows the data set. Figures (b) and (c) show the two results that SCO-RANSAC generated for different runs. Figure (d) shows the guaranteed result after adding the post processing step.

Chapter 3

Applications in Computer Vision

3.1 Fitting Multiple Affine Models

In this section, we apply SCO-RANSAC to fitting more complex parametric models, with more parameters and different fitting energies. We use SCO-RANSAC to estimate affine transformations induced by planes in the context of rectified narrow-base stereo. In this case SCO-RANSAC can be compared to an earlier work of Birchfield and Tomasi (BT) [4] who also used MRF-based regularization for fitting affine models.

3.1.1 Estimating Planes in Narrow-Based Stereo

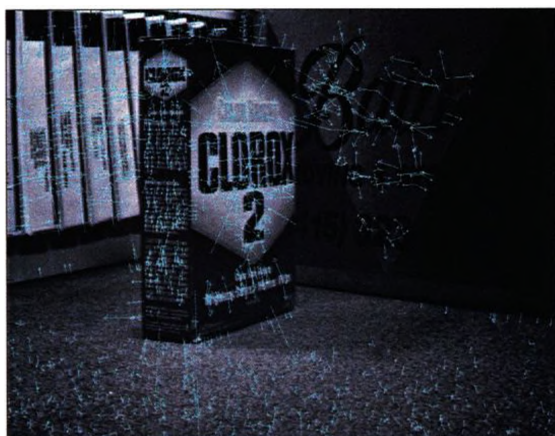
We use SIFT [28] features as points of interest, since they are scale- and rotation- invariant. SIFT features are also partially invariant to illumination¹ and 3D camera viewpoint changes. Matches between pairs of points in the two images were found using exhaustive search² along the corresponding scan line (see Fig. 3.1). When fitting affine models each *matched* feature is considered a data point.

The neighborhood graph was obtained by computing the Delaunay triangulation of the matched features of one of the images (the right image). We used constant discontinuity penalties w_{pq} for each pair of “neighboring” data points. In the synthetic examples described in chapter 2, the original data points of each model were perturbed with a random number (based on a Gaussian distribution with zero mean). Thus, it was legitimate to assume that the closer a pair of neighboring data points, the more likely that they belonged to the same model. Such an assumption could not be guaranteed to hold for the detected SIFT features.

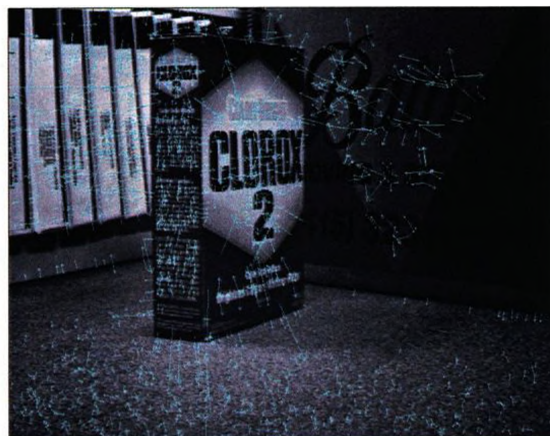
We will use the notation (x_l, y_l) , (x_r, y_r) to describe the coordinates of an image-matched feature on the left image p_l and right image p_r , respectively. Also, the symbol p will denote a pair of matching features (x_l, y_l, x_r, y_r) . A planar homography has only three degrees of freedom in the case of rectified images, as the epipole $\mathbf{e} = [1 \ 0 \ 0]^T$ is at infinity. In turn, the fundamental matrix could be formulated as ($F = [\mathbf{e}]_x$); we use the notation $[\mathbf{e}]_x$ to describe the skew-symmetric matrix of the vector \mathbf{e} . Since a planar

¹A low-contrast image can be automatically detected and adjusted. This is beyond the scope of the current topic.

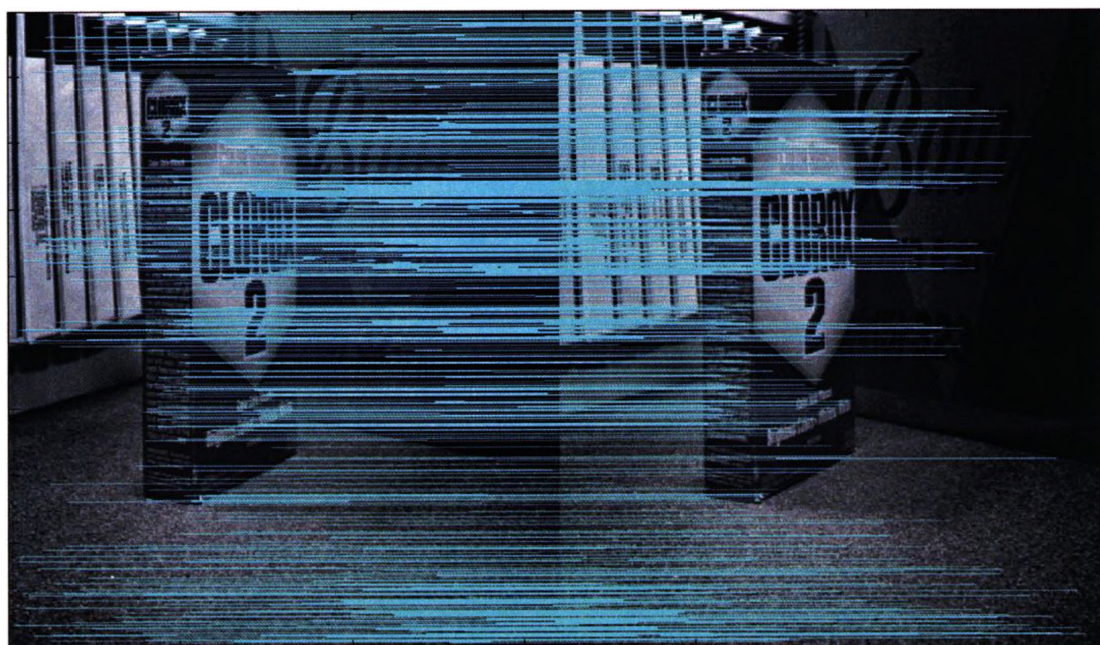
²In principle, it is possible to replace the exhaustive search with “smarter” methods, as in [30, 2].



(a) Left image



(b) Right image



(c) Matched features between images

Figure 3.1: Figures (a) and (b) show the SIFT matches found on the left and right images, respectively. Figure (c) shows the matched features between the two images.

homography must satisfy the following constraint: [14]: ($H^T F + F^T H = 0$). This will enforce the planar homography for the rectified stereo pair to be a special case of an affine transformation, with only three degrees of freedom:

$$A = \begin{pmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.1)$$

We will refer to this special affine transformation as the affine transformation for simplicity (see appendix A for complete proof).

The finite set of initial proposals A_0 is generated by randomly sampling three matching pairs and computing their corresponding model parameters. The result is obtained by applying SCO-RANSAC to minimize the following energy:

$$E(\mathbf{A}) = \sum_p ||p - A_p|| + \lambda \sum_{(p,q) \in \mathcal{N}} \delta(A_p \neq A_q) \quad (3.2)$$

where $\mathbf{A} = \{A_p | p \in P\}$ is an assignment of affine models to data points p . The following section considers using two different energies for fitting an affine transformation induced by a plane.

3.1.2 Geometric Fit Measures

3.1.2.1 Horizontal Shift

In our case based on (3.1), an affine model requires only three matching pairs to be uniquely identified. One way to measure the geometric error between a matching pair of points p and a given model A is by considering the horizontal shift between p_r and the mapping of p_l by A onto the right image (see Fig. 3.2)

$$||p - A|| = (ax_l + by_l + c - x_r)^2, \quad (3.3)$$

which is the horizontal shift (along the epipolar line) between p_r and Ap_l .

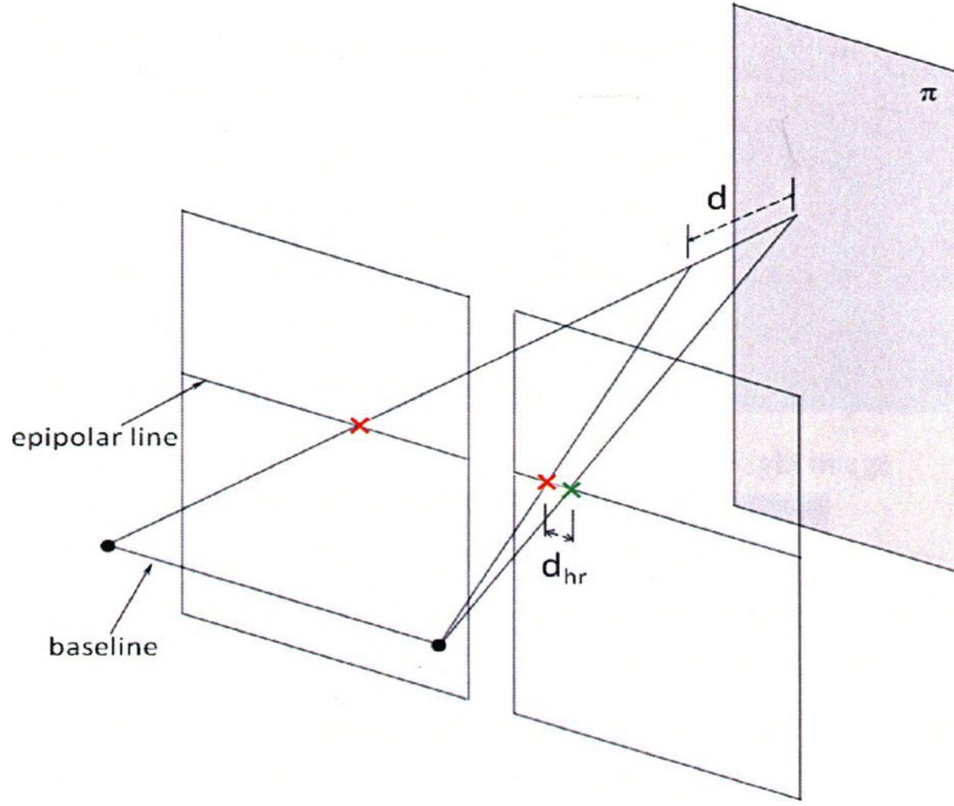
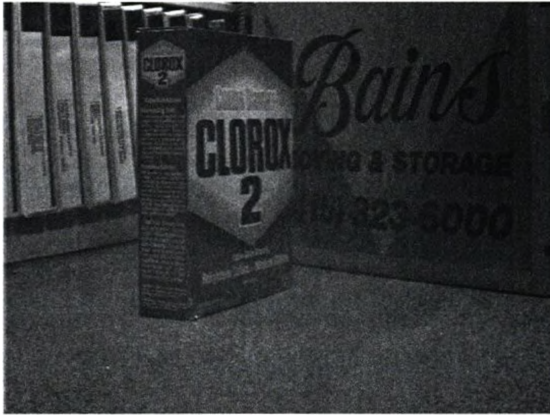
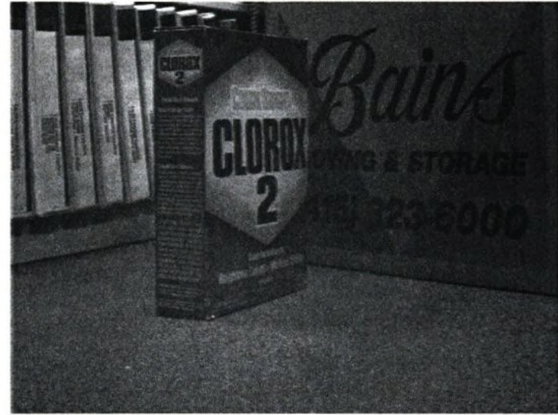


Figure 3.2: This figure illustrates the horizontal shift error d_{hr} . The distance between two triangulated points is denoted by d . The first triangulated point is obtained from p_l and p_r (shown in red). The second triangulated point is obtained from p_l and the mapping of p_l onto the second image by A_π . A_π is the affine/model induced by the plane π . The image of d onto the second image is denoted by d_{hr} .

Figure 3.3 shows SCO-RANSAC result for fitting multiple-affine models. The finite set of initial proposals A_0 was generated by randomly sampling three matching pairs and computing their corresponding model parameters using (3.3). The result is obtained by applying SCO-RANSAC to minimize the energy (3.2).



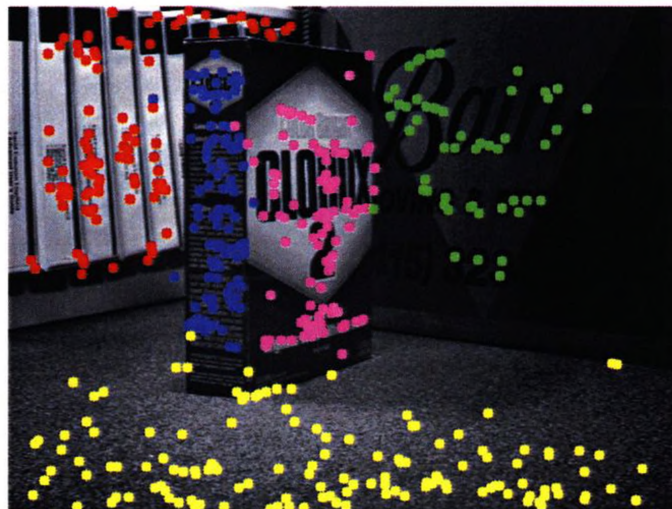
(a) Left image



(b) Right image



(c) SCO-RANSAC result using horizontal shift (3.3).



(c) SCO-RANSAC result using weighted shift (3.6).

Figure 3.3: Figures (a) and (b) show the original stereo pairs obtained by BT [4]. Figures (c) and (d) show SCO-RANSAC result for using the horizontal shift error (3.3) and the weighted shift error (3.6).

3.1.2.2 Weighted Shift

As can be seen from the results in Figure 3.3(c), the horizontal shift error (3.3) worked well for all the vertical planes in the scene, but it split the ground plane into two separate planes. The problem with the horizontal shift error (3.3) is illustrated in Figure 3.4. The red point is the triangulation of a matching pair of points. π_1 is the plane induced by the estimated affine transformation that minimizes the horizontal shift error. In this case, if there was no error in localizing the image features, the red point would lie on π_1 . π_2 is obtained by rotating π_1 around the shown axis³. As you can see, the horizontal shift error is the same for π_1 and π_2 , even though the perpendicular distance between the red point and π_2 is smaller than that between the red point and π_1 . Thus, the horizontal shift error is not a fair error estimate. It significantly overestimates the distance between a triangulated point and non-vertical planes in the 3D scene.

We propose using a fairer geometric error measure, we call “weighted shift”. The weighted shift error weights down the horizontal shift error for non-vertical planes. This error is derived from the fact that allowing some vertical shift for both corresponding points p_l and p_r to a different epipolar line adds an additional degree of freedom in computing the shortest distance to the perfect alignment with the affine model induced by a 3D world plane. This significantly reduces the error for non-vertical planes while not changing the error for vertical planes.

In order to derive the weighted shift error, we assume that a line belongs to a 3D titled plane and is mapped as a vertical line l on the left image, and thus as a tilted line l' on the right image [see Fig. 3.5(a)]. Figure 3.5(b) shows two corresponding points $p_l (x_l, y)$ and $p_r (x_r, y)$ (in red) and the mapping of p_l by A (in green) on the epipolar line y ; thus, the horizontal shift error is

$$E_{hr} = (ax_l + by_l + c - x_r)^2 = d_{hr}^2.$$

Shifting the epipolar line by Δy will cause the p_l and p_r to be shifted to $\tilde{p}_l (x_l, y + \Delta y)$ and $\tilde{p}_r (x_r, y + \Delta y)$, respectively. Also, \tilde{p}_l will be mapped to the blue point on l' by A .

³ The axis of rotation is line of intersection between π_1 and the the epipolar plane of p_r and p_l

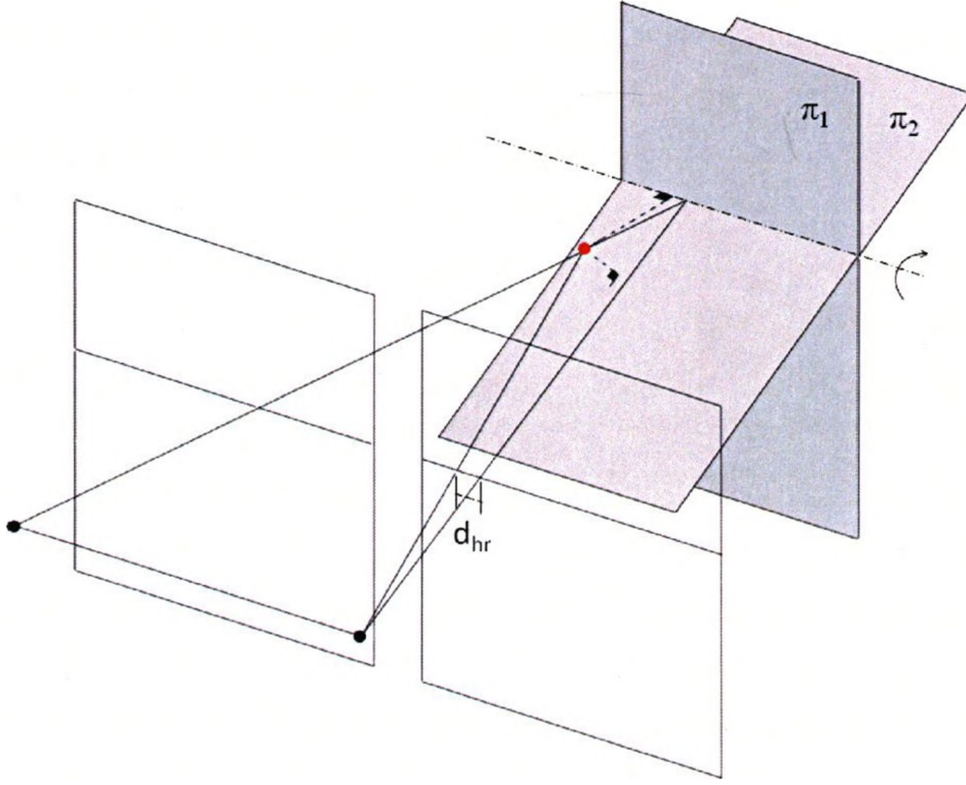


Figure 3.4: That figure illustrates that the horizontal shift error 3.3 does not take the slope of a plane into account. Obviously, the red point is closer to plane π_2 , even though the horizontal shift error d_{hr} is the same for both π_1 and π_2 .

The coordinates of the blue point are $(x_r + \Delta x, y + \Delta y)$ by construction, and $(ax_l + (y + \Delta y)b + c, y + \Delta y)$ as the mapping of \tilde{p}_l by A. Thus, we can derive the following relation between Δx and Δy

$$\begin{aligned}
 x_r + \Delta x &= ax_l + (y + \Delta y)b + c \\
 \Delta x &= ax_l + by + c - x_r + b\Delta y \\
 \Delta x &= d_{hr} + b\Delta y.
 \end{aligned} \tag{3.4}$$

The weighted shift error that we propose minimizes the squared distance between p_r (x_r, y) and the closest possible point that lies on l' (blue point). This error allows the epipolar line (of the matched pair) to shift vertically⁴. The squared distance between p_r (x_r, y) and the blue point is $(\Delta x^2 + \Delta y^2)$ by construction. Minimizing the squared distance

⁴The horizontal shift error is a special case of the weighted shift ($\Delta y = 0$).

subject to the constraint (3.4) will lead to the following solution:

$$\begin{aligned}\Delta x &= d_{hr} \frac{1}{1+b^2}, \\ \Delta y &= d_{hr} \frac{-b}{1+b^2},\end{aligned}$$

and thus the weighted shift error is

$$\begin{aligned}E_{whr} &= \Delta x^2 + \Delta y^2 \\ &= d_{hr}^2 \frac{1}{1+b^2} \\ &= E_{hr} \frac{1}{1+b^2} \\ &= \frac{(ax_l + by_l + c - x_r)^2}{1+b^2}.\end{aligned}$$

Finally, a weighting factor w can be introduced to remove the dependence of the weighting factor $(1/(1+b^2))$ on the coordinate system used. As, the value of b is dependent upon the coordinate system used:

$$E_{whr} = \frac{(ax_l + by_l + c - x_r)^2}{1 + w \cdot b^2}. \quad (3.5)$$

In summary, the weighted horizontal shift error weights down the horizontal shift error for non-vertical planes ($b > 0$) while not changing the error for vertical planes ($b = 0$).

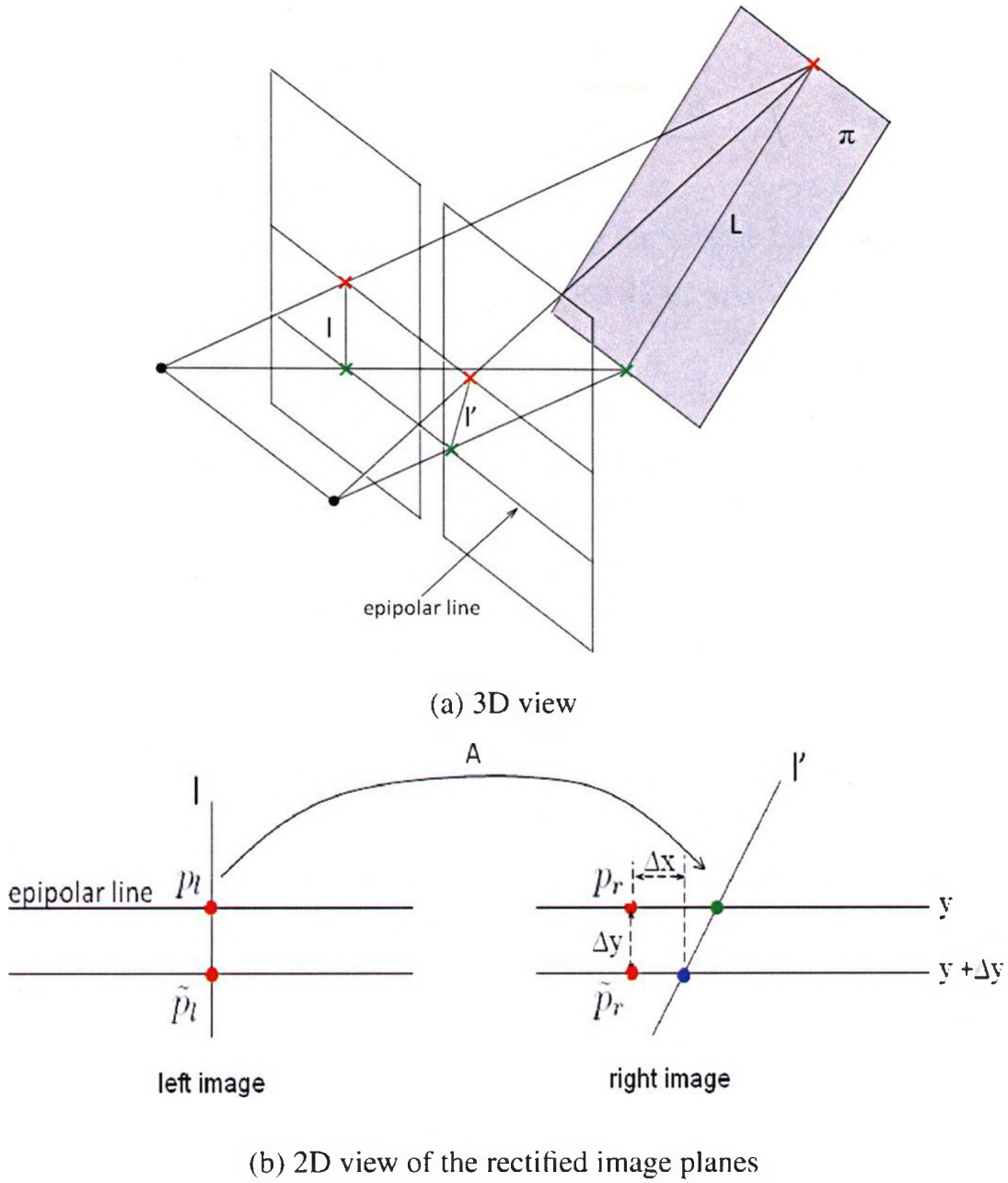


Figure 3.5: This figure illustrates the effect of allowing the epipolar line of matched pair of points p_l, p_r to shift. Figure (a) shows that line L belongs to the plane π and it is mapped as l and l' on the left and right images, respectively. Figure (b) shows the 2D view of the two rectified image planes.

The weighted shift error (3.5) is a nonlinear error. The initial solution required by any nonlinear least squares minimization technique could be the models obtained by SCORANSAC for minimizing the horizontal shift error. The geometric error between a match-

ing pair of points p and a given model A in this case is

$$\|p - A\| = \frac{(ax_l + by_l + c - x_r)^2}{1 + w \cdot b^2}. \quad (3.6)$$

Figure 3.3(d) shows the results obtained while using the weighted shift error. As can be seen using the appropriate geometric error, SCO-RANSAC was able to find the ground plane.

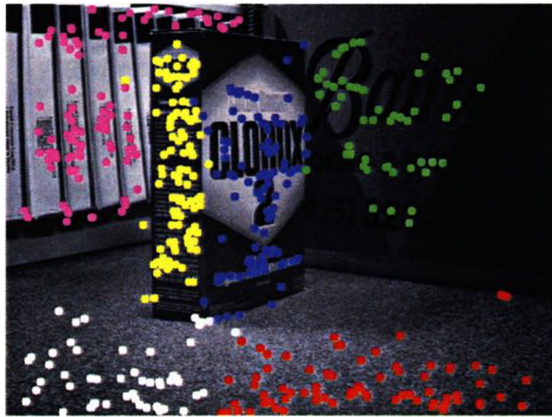
3.1.3 SCO-RANSAC vs. Birchfield and Tomasi

Our approach to sparse stereo differs from the BT dense stereo method in many ways. The most significant difference is that our framework for estimating models automatically generates initial proposals by random sampling, as in RANSAC-based methods. In contrast, BT [4] compute initial estimates of affine models from a disparity map generated by another dense stereo method [6].

There are also significant differences in our formulations of stereo. Our model-fitting is based on geometric alignment of matched features, as common in sparse stereo. In contrast, BT [4] used a photoconsistency-based measure, typical for dense stereo. The method in [4] is limited to disparity maps and narrow-based stereo, as it does not handle large occlusions common with wide-based stereo.

Their focus on narrow-based stereo justifies the use of affine models in [4]. In contrast, our method applies to narrow – and wide – based stereo and could be potentially extended to volumetric multi-view reconstruction. Also, the dense stereo method in [4] is bound to regular grids, while our method can be used with any triangulation.

Figure 3.6 compares affine model fitting results generated by BT [4] and the two results generated by SCO-RANSAC using different geometric error measures $\|p - A\|$. BT [4] use dense segmentation of pixels based on photoconsistency. This measure does not work well in textureless regions and they have to rely on intensity edges (static cues) to detect the boundaries between regions supporting different models. In contrast, SCO-RANSAC labels a sparse set of distinct features based on geometric (or algebraic) errors and spatial proximity.



(a) SCO-RANSAC
horizontal shift error



(b) SCO-RANSAC
weighted shift error



(c) BT [4] result

Figure 3.6: Comparisons of the results for the Clorox stereo pair [4]. Figures (a) and (b) show sparse inlier classification by SCO-RANSAC using different geometric fit measures. Figure (c) show dense pixel segmentation by BT [4] using photoconsistency. BT's method did not segment the Clorox box correctly, and it split the plane corresponding to the books.

In order to provide some quantitative comparison between the affine models generated by BT [4] and SCO-RANSAC, we found the “ground truth models” (see Fig. 3.7) by manually extracting point (and line) correspondences and computing the affine model for each plane except the “books” plane.

Assuming that two intersecting planes, π_1 and π_2 , are represented by the affine models A^{π_1} and A^{π_2} , then the homogeneous vector representing their line of intersection is defined as the first row of the matrix $(A^{\pi_1} - A^{\pi_2})$. Therefore, such lines can be computed from the models estimated by either BT⁵ [4] or SCO-RANSAC. Table 3.1 compares the geometric accuracy of the computed lines of intersection using the models found by each method. Figures 3.9, 3.8, and 3.10 show the computed lines of intersection.

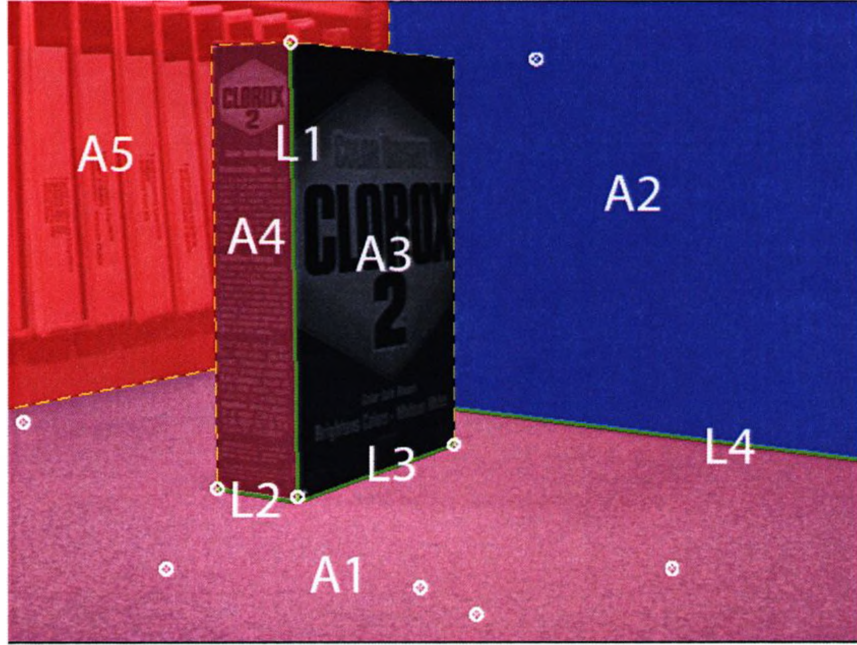


Figure 3.7: We computed the ground truth affine models for the planes A_1, \dots, A_4 using 10 manually matched features (shown in white) and a line L_4 . The lines L_1, \dots, L_4 (shown in green) are the lines of intersection computed using the ground truth affine models. The computed affine model for the ground plane was up to 0.25 pixel accurate.

⁵BT used affine models with six degrees of freedom, although they mentioned that for rectified images the motion of a plane could be described only by three parameters only. In computing the lines of intersection using their models we approximated the second row of their affine models to be $[0 \ 1 \ 0]$. For all the affine models found by BT [4], the second row was approximately $[0 \ 1 \ 0]$, except for the ground plane.

Line	BT [4]	SCO-RANSAC (3.3)	SCO-RANSAC (3.6)
L_1	34.55	8.80	9.46
L_2	16.83	4.82	3.66
L_3	5.56	13.27	6.26
L_4	5.99	4.46	12.05
Total	62.94	31.37	31.45

Table 3.1: Show the geometric errors for lines where affine models intersect. Errors were computed as the sum of distances between the ground truth line segment corners and the computed lines.



Figure 3.8: SCO-RANSAC using the horizontal shift error. Lines were computed for all pairs of intersecting planes (affine models).



Figure 3.9: BT [4] using photoconsistency measure. Lines were computed for all pairs of intersecting planes (affine models).



Figure 3.10: SCO-RANSAC using the weighted shift error. Lines were computed for all pairs of intersecting planes (affine models).

3.2 Fitting Multiple Homographies

In this section, we apply SCO-RANSAC to fitting models with a larger number of parameters ($n = 8$), and a nonlinear geometric fit measure. We use SCO-RANSAC to estimate multiple homographies in the context of uncalibrated wide-based stereo. Finally, we compare SCO-RANSAC results to those obtained by Sequential RANSAC [39], multi-RANSAC [45], and the Kanazawa and Kawakami method [24].

3.2.1 Estimating Planes in Uncalibrated Wide-Based Stereo

The matched features and the neighborhood graph in this case are obtained exactly as described in the case of fitting affine models, except in this case we do not assume calibration. Thus, each feature on the left image is tested against all the features on the right image to find the match. This leads to a larger number of outliers that will cause gaps. We will show how to deal with these gaps in section 3.2.2. The gaps issue was referred to in the introduction section 1.1 as a case of spatially disconnected inliers.

One way to measure the geometric error $\|p - H\|$ between a matching pair of points p and a given homography H is by using the symmetric transfer error (STE) [19]. We generate our finite set of initial model proposals H_0 by randomly sampling four matching pairs at a time. Then we compute the model parameters as described in [19] by minimizing the nonlinear least squares of the STEs using Levenberg-Marquard technique, which is covered in appendix B. The energy function to be minimized using SCO-RANSAC is

$$E(\mathbf{H}) = \sum_p \|p - H_p\| + \lambda \sum_{(p,q) \in \mathcal{N}} \delta(H_p \neq H_q)$$

where $\mathbf{H} = \{H|p \in P\}$ is an assignment of models to data points p .

3.2.2 Removing Outliers and Retriangulating Holes

In the case of line fitting, SCO-RANSAC uses a 2D triangulation as the neighborhood graph for the 2D data points, while in case of stereo images SCO-RANSAC uses a 2D

triangulation of the matched features even though the matched features impede 3D information (see Fig. 3.11). Thus, by using a 2D triangulation, outliers will cause gaps between the inliers of a model. When the number of outliers increases, it becomes possible for the introduced gaps to split the inliers of a model into spatially disconnected sets, which violates our assumption that the inliers of a model are spatially coherent.

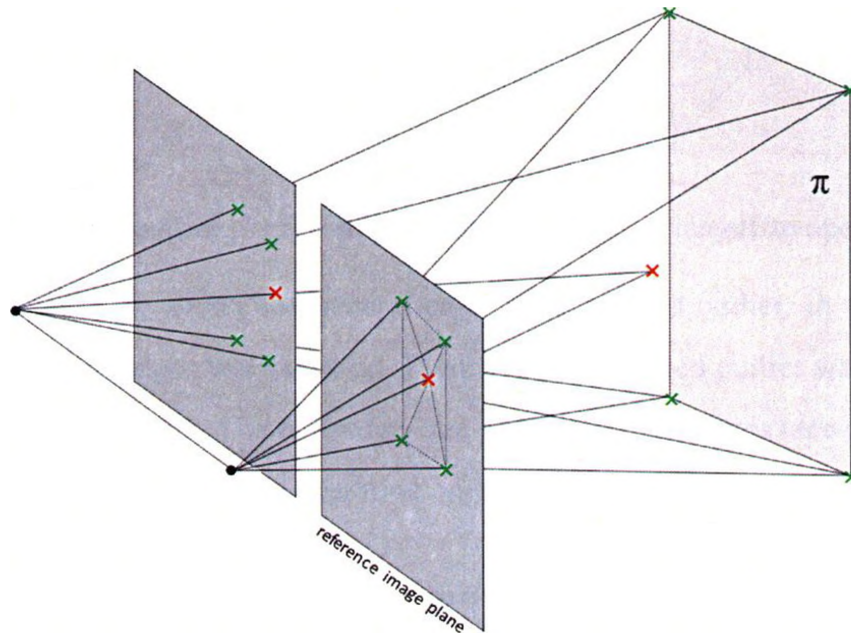


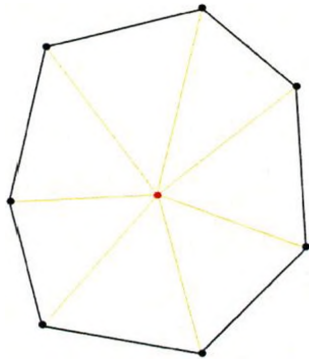
Figure 3.11: This figure shows the gap that an outlier for the plane π will introduce if points are triangulated in one of the image planes instead of 3D.

Outliers occur in two forms: isolated outliers (random false matches) or clusters (e.g., a symmetry in the scene might lead to multiple false matches that are spatially coherent. This will form a cluster of outliers).

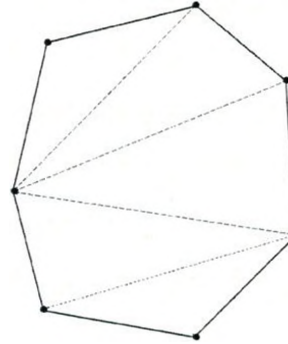
Definition 4 *An isolated outlier is said to exist, if after an iteration of SCO-RANSAC there exists a model such that its CS contains only one data point. This data point is an outlier as no model with large data support fits it.*

In practice, we found that the isolated outliers are more likely to occur than the clustered case. Thus, we will modify the SCO-RANSAC algorithm to solve the isolated outliers case and add a post processing step to avoid having a set of clustered outliers as a legitimate model. In order to cope with the isolated outliers, after each iteration we remove the

isolated outliers from the Delaunay triangulation and then retriangulate the resulting holes (see Fig. 3.12). The resulting global triangulation remains a delaunay triangulation [13].



(a) Isolated outlier



(b) After the remove and retriangulate operation

Figure 3.12: Figure (a) shows the general case of an isolated outlier, in which we only know that the yellow edges were severed. Figure (b) the isolated outlier was removed and the hole was retriangulated. The new edges are shown as dotted lines (see appendix C for further details on the effect of this operation on the energy).

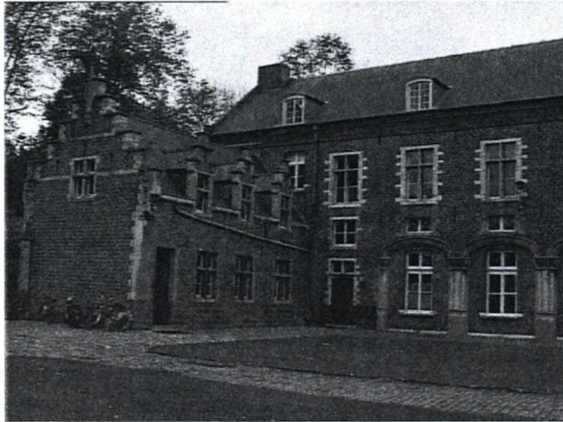
Clustered outliers can also occur due to two other cases: The first case occurs when a model exists but its CS is not large enough to efficiently estimate the model's parameters. In this case, the inliers for that model should be treated as outliers. The second case occurs when the CS of a model do not follow a certain criteria required in order to estimate the model (e.g., the inliers of a homography should not be collinear). Following Santos-Pereira and Pires's [34] criteria for detecting such clusters, we used a post processing step that removes any model with a CS smaller than $(2n + 2)$, where n is the number of the model parameters. Their argument for using this criterion is that for a number of data points smaller than $(2n + 2)$, the covariance matrix⁶ estimates become unreliable.

3.2.3 Comparisons and Results

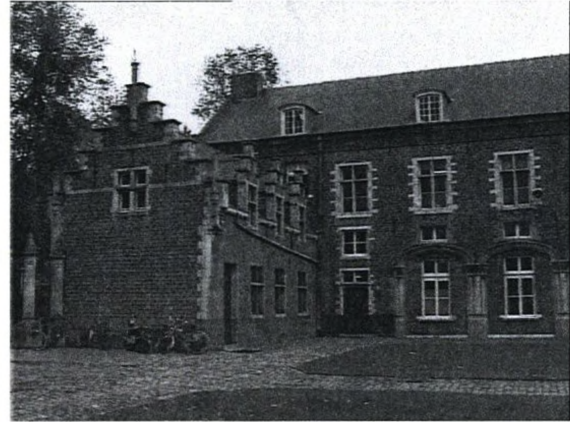
Only 500 initial models were used in the examples shown in Figures 3.14 and 3.13. SCORANSAC converged after three iterations in both examples. In these two examples we will

⁶The covariance matrix is used to estimate a confidence limit of the fit.

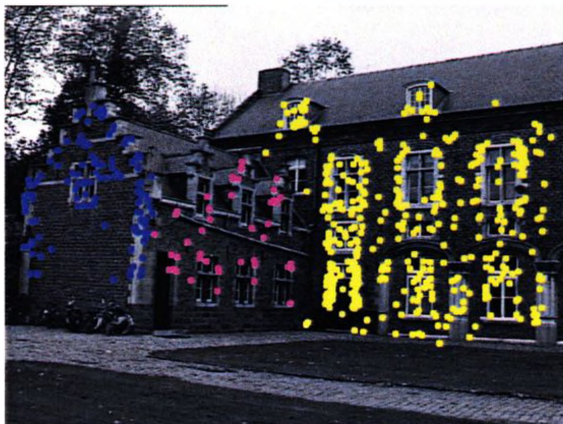
show the inlier classification on the left and right images. In the rest of the results and comparisons we will show the inlier classification on only one of the images.



(a) Left image



(b) Right image



(c) Inliers classification
shown on the left image



(d) Inliers classification
shown on the right image

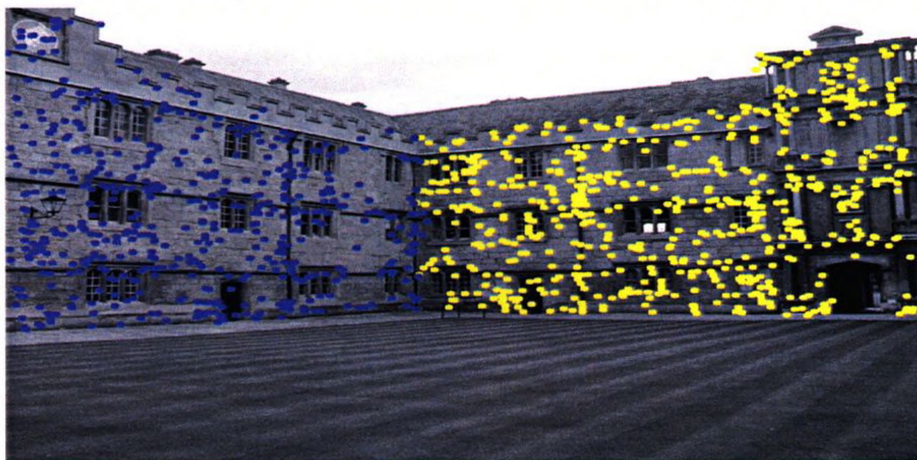
Figure 3.13: SCO-RANSAC was able to find three planes. The original images are taken from Marc Pollefeys's Leuven castle data set.



(a) Left image



(b) Right image



(c) Inliers classification shown on the left image



(d) Inliers classification shown on the right image

Figure 3.14: SCO-RANSAC was able to find two planes. The original images are from VGG (Oxford) Merton College. SCO-RANSAC was able to accurately classify the inliers. Notice the place where the two planes intersect.

3.2.3.1 SCO-RANSAC vs. Sequential RANSAC Methods

In the example shown in Figure 3.15(e) SCO-RANSAC was able to identify three planes using 500 initial labels. SCO-RANSAC found the three dominate planes: the staircase and the two vertical planes shown in blue, yellow, and violet, respectively. SCO-RANSAC converged after five iterations.

Sequential RANSAC [39] required 817 random samples and found five planes. Two of them were the vertical planes while the remaining three were an over fit for the staircase plane. The over fitting is due to Sequential RANSAC sensitivity towards the threshold parameter. Using higher threshold would merge the three staircase planes into one plane, also that will cause one of the two vertical planes to grab more inliers on the account of the other vertical plane.

Kanazawa and Kawakami [24] proposed a robust method for detecting local planar regions in a scene with an uncalibrated stereo. Their method is a variation of Sequential RANSAC. They use the distributions for each feature point, defined by the distances between the point and the other points, to generate more appropriate initial models, not just random models. They first sample a matched feature uniformly and then the next matched feature gets sampled based on distribution of the previously sampled matched feature. By doing this, they increase the probability of sampling four matched features that belong to the same plane. Their method does not guarantee spatially coherent regions as mentioned in [24] (see Fig. 3.15(d)). The Kanazawa and Kawakami [24] method found three planes. The plane shown in red was not spatially coherent.



(a) Left image



(b) Right image

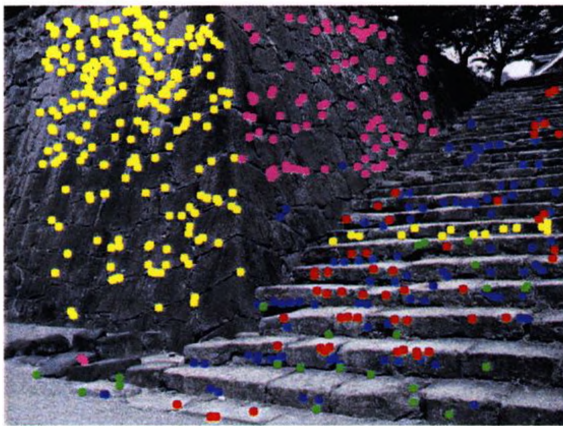
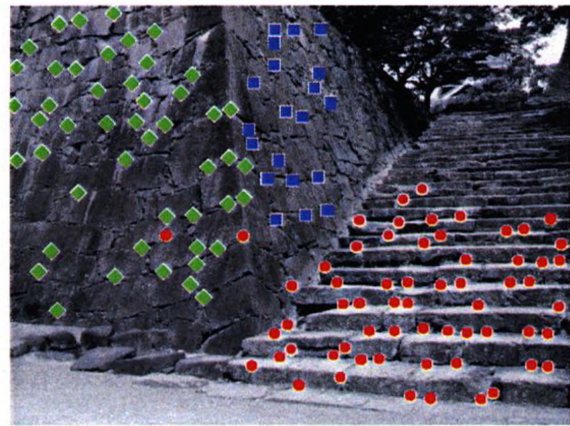
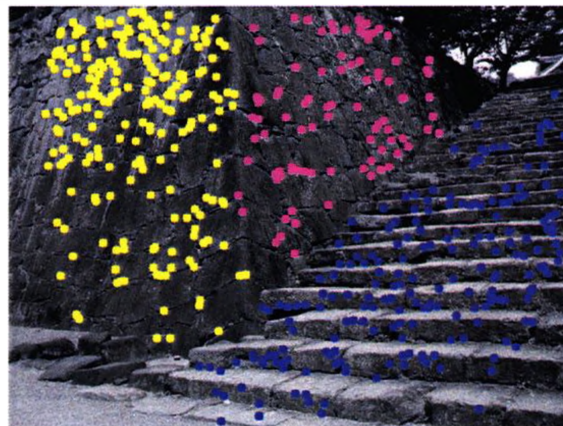
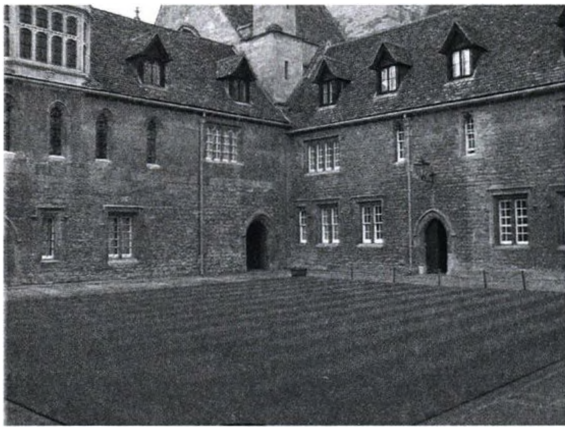
(c) Sequential RANSAC
using SIFT(d) Kanazawa and Kawakami [24]
using Harris corners(e) SCO-RANSAC result
using SIFT

Figure 3.15: This figure compares Sequential RANSAC and Kanazawa and Kawakami method to SCO-RANSAC. The original stereo pair is from Kanazawa and Kawakami [24].

3.2.3.2 SCO-RANSAC vs. Multi-RANSAC

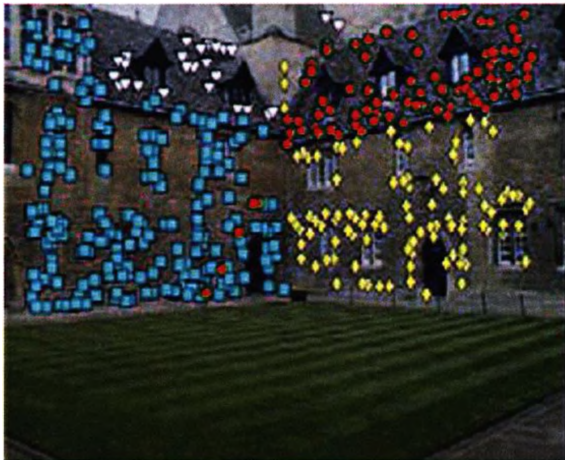
In the example, shown in Figure 3.16 SCO-RANSAC was able to identify eight planes. Unlike multiRANSAC [45], SCO-RANSAC does not require the prior knowledge of the number of planes. Also SCO-RANSAC produces spatially coherent set of inliers for each plane as compared to multiRANSAC. In [45] multiRANSAC required 46,416 random samples in order to find just four planes. On the other hand SCO-RANSAC used only 900 initial random samples (2% of the number required by multiRANSAC) and found more planes than multiRANSAC. SCO-RANSAC required only three iterations to converge.



(a) Left image



(b) Right image



(c) MultiRANSAC [45]



(d) SCO-RANSAC

Figure 3.16: Comparing multiRANSAC and SCO-RANSAC using stereo images from VGG (Oxford) Merton College I.

3.2.3.3 Scenes with Large Number of Models

In this section, we only computed results for SCO-RANSAC, as the other RANSAC-based methods would have required a gigantic number of initial models, especially multiRANSAC. In the example shown in Figure 3.2.3.3 SCO-RANSAC was able to identify eight planes. Only 2,800 initial labels were used, and SCO-RANSAC converged after four iterations. Notice how planes with varying numbers of inliers were easily identified by SCO-RANSAC. For example, the roof of the building on the left (shown in light brown) was supported by very few inliers compared to its two neighboring large planes (shown in blue and purple).

Figure 3.18 shows Raglan Castle Tower. Using 5,500 initial labels and only four iterations to convergence SCO-RANSAC was able to identify 15 planes. Identifying this number of models was not archived before. The maximum number of identified planes by the RANSAC-based methods in [39], [24], and [45] was 3, 3, and 4, respectively. The use of a relatively large number of initial labels allowed SCO-RANSAC to identify very small structures. Notice how SCO-RANSAC was able to identify very small planes like those that are marked by red and green rectangles in Figure 3.18(a). SCO-RANSAC actually was able to identify smaller coherent structures but these were filtered out in the post processing step.

The first time we used SCO-RANSAC to find the models for this stereo pair, we thought that the result was over segmented, based on the assumption that there should be only four planes —a plane for each vertical segment of the tower. Later on, we discovered from looking at another picture shown in Figure 3.19 that each floor shifts to the inside by one step from the floor beneath it. Simply, SCO-RANSAC was able to see what our eyes failed to see.



(a) Left image

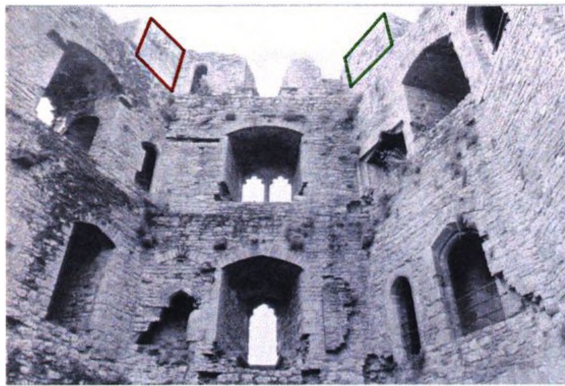


(b) Right image



(c) SCO-RANSAC result

Figure 3.17: Stereo images from VGG (Oxford) Merton College III.



(a) Left image



(b) Right image



(c) SCO-RANSAC result

Figure 3.18: In this example from VGG (Oxford), we used the same color more than once to represent different planes. Only spatially connected planes are shown in different colors. Figure (a) shows red and green rectangles, which are drawn to facilitate their referencing in the text.



Figure 3.19: Image of Raglan Castle tower found on Flickr. Anonymous photographer.

3.3 Conclusion

We conclude that our new general fitting algorithm is robust to outliers and noise in the data. SCO-RANSAC proved to be very efficient in fitting multiple lines compared to the other RANSAC-based methods. By using a spatial prior, SCO-RANSAC eliminated the need for a threshold parameter to classify the inliers. Empirically SCO-RANSAC requires far fewer hypothetical models than the other methods. As shown earlier, SCO-RANSAC required only 2% of the hypotheses required by multiRANSAC, and it found a larger number of models than multiRANSAC. SCO-RANSAC's complexity was shown to be comparable to the complexities of the other methods described.

3.4 Extensions

We aim to extend SCO-RANSAC to volumetric multi-view reconstruction, and use other triangulations as well. We also intend to generalize the isolated outlier removal technique to be able to handle clustered outliers. By doing this, the weak model filtering post process will be part of the SCO-RANSAC process. We also intend to investigate the possibility of clustering higher dimensional data by using their projection on lower dimensional subspaces. For example, in the case of stereo vision, we were able to fit planes in 3D (by computing their affine transformation) by using the information incorporated in two 2D images. We hope to be able to cluster 4D points by using their projections on multiple 3D subspaces.

Using a piecewise-smoothness model other than piecewise-constant (pott's) model would be more helpful to identify and fit close models (close in the model parameter space). We aim to investigate the use of SCO-RANSAC in detecting multiple motions. One previous method[42] for estimating multiple motion used regularization in the final step (dense segmentation) only not in the motion detection step. Unlike that method SCO-RANSAC uses regularization in model estimation.

Appendix A

Plane-to-Plane Transformation in Rectified Images

Assuming that the projection matrices for the two views are

$$\mathbf{P} = \left[I \mid \mathbf{0} \right] \quad \text{and} \quad \mathbf{P}' = \left[M \mid \mathbf{a} \right],$$

and the plane is defined by π with $\pi = (\mathbf{v}^T, 1)^T$, then the homography H_π induced by that plane is

$$H_\pi = A - \mathbf{a}\mathbf{v}^T.$$

The complete proof of the formulation can be found in [19].

Definition 5 *Let H be a homography from the first image to the second image. We say that this homography is a **planar homography** if and only if there exists a plane π such that $H = H_\pi$.*

Theorem 1 *Given a pair of images and the corresponding fundamental matrix F , let there be a homography from the first to the second image plane represented by the matrix H . The following conditions are equivalent:*

1. *The homography H is a planar homography.*
2. $H^T F + F^T H = 0$.

For rectified images, the epipole e is at infinity and the fundamental matrix F can be formulated as $F = [e]_x$. We use the notation $[e]_x$ to describe the skew-symmetric matrix of the vector e :

$$F = [e]_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}.$$

One possible way to restrict a homography to be planar is to enforce the second condition in Theorem (1), rather than the first one, as they are both equivalent. For the complete proof of that theorem (1) refer to [14]. A homography must satisfy the constraint $H^T F + F^T H = 0$ [14] in order to be planar. Enforcing this constraint implies the following :

$$\begin{aligned} H^T F + F^T H &= 0 \\ \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}^T \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} &= 0 \\ \begin{bmatrix} 0 & h_7 & -h_4 \\ h_7 & 2h_8 & h_9 - h_5 \\ -h_4 & h_9 - h_5 & -2h_6 \end{bmatrix} &= 0, \end{aligned}$$

which implies the following :

$$h_7 = 0, h_4 = 0, h_8 = 0, h_5 = h_9, \text{ and } h_6 = 0.$$

Since H is defined up to scale then we can enforce that $h_9 = 1$ without loss of generality¹. Finally we arrive at the following conclusion; a planar homography H , in the case of

¹In the case of a narrow-based stereo, a finite point will always be mapped to a finite point, and a point at infinity will be mapped to a point at infinity; thus, it is safe to assume that $h_9 \neq 0$.

rectified images, can be parameterized as follows:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Since this parametrization is a special case of the affine transformation, we will refer to it as an affine transformation (and denote it by the matrix A) for simplicity.

$$A = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

where a, b , and c are the three parameters that define the affine model.

Appendix B

Levenberg-Marquard

The Levenberg-Marquard algorithm [12] is a modification of Newton's algorithm, which can be used to find local minima of a nonlinear function. Newton's algorithm uses the first and second derivatives of the function, and thus performs better than the well-known gradient descent, which uses only the first derivative. Given a starting point, Newton's algorithm constructs a quadratic approximation for the objective function that matches the first and second derivative values at that point. It then minimizes the quadratic function instead of the original objective function. Finally, it uses the minimizer of the approximate function as the starting point for the following step and this procedure is repeated until convergence.

Assuming that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function and that it is twice continuously differentiable, we can obtain the quadratic approximation q for f at a certain point \mathbf{x}^k by using the Taylor expansion of f and neglecting terms of order three and higher as follows:

$$f(\mathbf{x}) \approx f(\mathbf{x}^k) + (\mathbf{x} - \mathbf{x}^k)^T \mathbf{g}^k + \frac{1}{2}(\mathbf{x} - \mathbf{x}^k)^T \mathbf{F}^k (\mathbf{x} - \mathbf{x}^k) = q(\mathbf{x})$$

where \mathbf{g}^k and \mathbf{F}^k denote the first and second derivatives of the function f at \mathbf{x}^k .

Theorem 2 First Order Necessary Condition. *Let Ω be a subset of \mathbb{R}^n , and f a real valued function on Ω . If \mathbf{x}^* is a local minimizer of f over Ω , then for any feasible direction \mathbf{d} at \mathbf{x}^* , we have*

$$\mathbf{d}^T \nabla f(\mathbf{x}^*) \geq 0.$$

Applying the first order necessary condition on q , we deduce the following:

$$\nabla q(\mathbf{x}) = \mathbf{g}^k + \mathbf{F}^k(\mathbf{x} - \mathbf{x}^k) = 0.$$

And, if $\mathbf{F}^k > 0$, then q archives minimum at

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\mathbf{F}^k)^{-1} \mathbf{g}^k.$$

If the Hessian matrix \mathbf{F}^k is not positive definite ($\mathbf{F}^k < 0$), then the direction obtained by $(-\mathbf{F}^k)^{-1} \mathbf{g}^k$ may not point in the descent direction. The Levenberg-Marquard technique simply augments the symmetric matrix \mathbf{F} to be $(\mathbf{F} + \mu_k \mathbf{I})$, where $\mu_k \geq 0$ guarantees that the eigenvalues of $(\mathbf{F} + \mu_k \mathbf{I})$ are all positive, and thus $(\mathbf{F} + \mu_k \mathbf{I})$ becomes positive definite. If μ_k is sufficiently large, then the search direction $(-\mathbf{F}^k + \mu_k \mathbf{I})^{-1} \mathbf{g}^k$ will always point in a decent direction. For the detailed derivation and comparisons between gradient descent, Newton's algorithm, and Levenberg-Marquard, refer to [12]. After introducing the Levenberg-Marquard modification the recurve formula becomes

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\mathbf{F}^k + \mu_k \mathbf{I})^{-1} \mathbf{g}^k.$$

For the case in which Levenberg-Marquard is used to solve nonlinear least-squares problem, this formula could be approximated by

$$\mathbf{x}^{k+1} = \mathbf{x}^k - ((\mathbf{J}^k)^T \mathbf{J}^k + \mu_k \mathbf{I})^{-1} (\mathbf{J}^k)^T \mathbf{r}^k,$$

where $\mathbf{r}_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, m$ are given functions, and the objective function to be minimized is $\sum_{i=1}^m (\mathbf{r}_i(\mathbf{x}))^2$ and \mathbf{J}^k is the Jacobian of \mathbf{r} at \mathbf{x}^k :

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial r_1}{\partial x_1}(\mathbf{x}) & \dots & \frac{\partial r_1}{\partial x_n}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial r_m}{\partial x_1}(\mathbf{x}) & \dots & \frac{\partial r_m}{\partial x_n}(\mathbf{x}) \end{bmatrix}. \quad (\text{B.1})$$

B.1 Minimizing The Symmetric Transfer Error Using LM

In this section we will use the LM technique for estimating a homography H that minimizes the symmetric transfer error, such that

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

and

$$H^{-1} = \begin{bmatrix} \hat{h}_1 & \hat{h}_2 & \hat{h}_3 \\ \hat{h}_4 & \hat{h}_5 & \hat{h}_6 \\ \hat{h}_7 & \hat{h}_8 & \hat{h}_9 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}^{-1},$$

and the symmetric transfer error $\mathbf{r} : \mathbb{R}^9 \rightarrow \mathbb{R}^4$, assuming that p_i and p'_i are two matching pairs:

$$\begin{bmatrix} \mathbf{r}_1(p'_i, p_i) \\ \mathbf{r}_2(p'_i, p_i) \\ \mathbf{r}_3(p'_i, p_i) \\ \mathbf{r}_4(p'_i, p_i) \end{bmatrix} = \begin{bmatrix} \frac{x_i h_1 + y_i h_2 + h_3}{x_i h_7 + y_i h_8 + h_9} - x'_i \\ \frac{x_i h_4 + y_i h_5 + h_6}{x_i h_7 + y_i h_8 + h_9} - y'_i \\ \frac{x'_i \hat{h}_1 + y'_i \hat{h}_2 + \hat{h}_3}{x'_i \hat{h}_7 + y'_i \hat{h}_8 + \hat{h}_9} - x_i \\ \frac{x'_i \hat{h}_4 + y'_i \hat{h}_5 + \hat{h}_6}{x'_i \hat{h}_7 + y'_i \hat{h}_8 + \hat{h}_9} - y_i \end{bmatrix}.$$

The LM technique requires an initial solution for H that can be obtained by solving for H , that minimizes the linear algebraic error:

$$\begin{aligned} P'_i &\simeq H P_i \\ P'_i \times H P_i &= 0 \\ [P'_i]_x H P_i &= 0 \end{aligned} \tag{B.2}$$

such that P_i is the homogeneous coordinate of p_i and $[v]_x$ represents the skew-symmetric matrix of the vector v . Given a set of n matching pairs, the system of equations in B.2 will generate two linearly independent equations for each pair. Thus, solving the following system of $2 \cdot n$ equations for $n \geq 4$ using SVD will result in the initial values of H that

minimize the algebraic error. The coordinates of the matching pairs must be normalized, as in [20], pre-conditioning the system equation to provide a better conditioned system of equations (refer to direct linear transform [19]).

Once we have an initial solution for H the following recursive formula of LM can be used to find H that minimizes the sum of squared transfer errors for all the n matching pairs:

$$H^{k+1} = H^k - ((\mathbf{J}^k)^T \mathbf{J}^k + \mu_k \mathbf{I})^{-1} (\mathbf{J}^k)^T \mathbf{r}^k, \quad (\text{B.3})$$

where

$$\mathbf{J} = \begin{bmatrix} \frac{\partial r_1}{\partial h_1}(p'_1, p_1) & \cdots & \frac{\partial r_1}{\partial h_9}(p'_1, p_1) \\ \frac{\partial r_2}{\partial h_1}(p'_1, p_1) & \cdots & \frac{\partial r_2}{\partial h_9}(p'_1, p_1) \\ \frac{\partial r_3}{\partial h_1}(p'_1, p_1) & \cdots & \frac{\partial r_3}{\partial h_9}(p'_1, p_1) \\ \frac{\partial r_4}{\partial h_1}(p'_1, p_1) & \cdots & \frac{\partial r_4}{\partial h_9}(p'_1, p_1) \\ \vdots & & \vdots \\ \frac{\partial r_1}{\partial h_1}(p'_n, p_n) & \cdots & \frac{\partial r_1}{\partial h_9}(p'_n, p_n) \\ \frac{\partial r_2}{\partial h_1}(p'_n, p_n) & \cdots & \frac{\partial r_2}{\partial h_9}(p'_n, p_n) \\ \frac{\partial r_3}{\partial h_1}(p'_n, p_n) & \cdots & \frac{\partial r_3}{\partial h_9}(p'_n, p_n) \\ \frac{\partial r_4}{\partial h_1}(p'_n, p_n) & \cdots & \frac{\partial r_4}{\partial h_9}(p'_n, p_n) \end{bmatrix}.$$

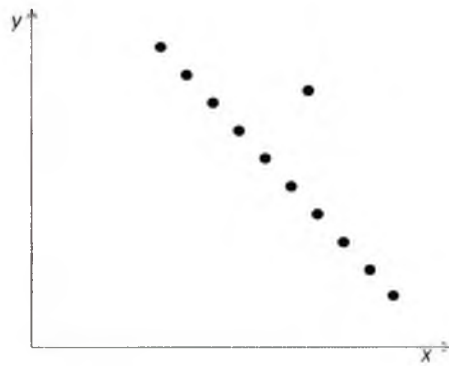
As mentioned in [19], a typical initial value for μ_k is 10^{-3} times the average of the diagonal elements of $\mathbf{J}^T \mathbf{J}$. If the new estimate leads to a decrease in the error, then the current μ_k is divided by a factor (typically 10) before the next iteration. On the other hand, if the current μ_k increases the error, then the μ_k is multiplied by the same factor. The process of searching for a μ_k value that minimizes the error constitutes one iteration of LM. The LM iterations are repeated until the energy either stops decreasing or the amount of reduction in the symmetric transfer error is too small.

Appendix C

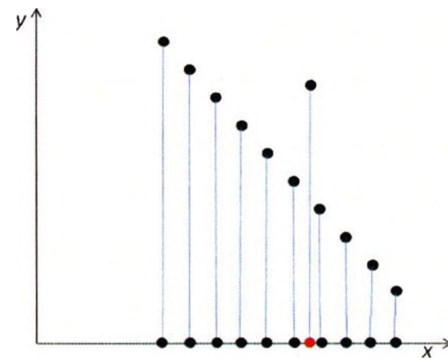
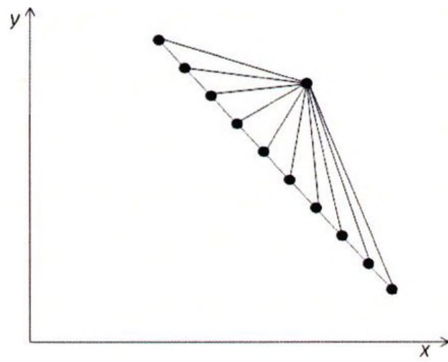
Outlier Gap in the Case of a 2D Line

In chapter 2, we demonstrated how to apply SCO-RANSAC to fit multiple lines [see Fig. C.1(a)]. In that case, we used the 2D triangulation of points [see Fig. C.1(c)]. In this appendix, we will show how to apply SCO-RANSAC to the same problem, while using a 1D triangulation. The 1D triangulation is obtained by first projecting the data points onto a subspace ($y = 0$) [see Fig. C.1(b)], and then triangulating the projected points [see Fig. C.1(d)]. By comparing Figures C.1 (c) and (d), one can easily see the amount of information about the spatial connectivity between the data points that is lost when using 1D triangulation.

Figures C.1 (e) and (f) show SCO-RANSAC results for 2D and 1D triangulation, respectively. As can be seen from these two results, the triangulation of points on 1D affects the inferred information about the spatial connectivity between the data points. In the case of 2D triangulation SCO-RANSAC found two models (one for the line and the other for the outlier), while in the case of 1D triangulation SCO-RANSAC found three models. As the outlier caused a gap that prevented the points of the other two models (shown in yellow and blue) from merging.



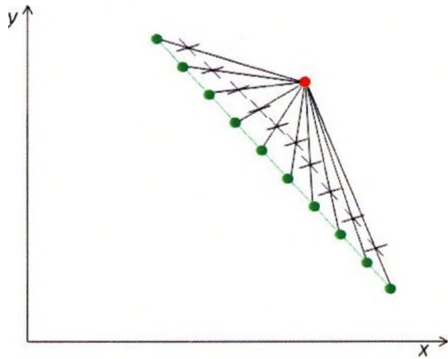
(a) 2D data points

(b) 2D points projected on a 1D subspace
($y = 0$)

(c) Triangulation on the 2D space



(d) Triangulation on the 1D subspace



(e) Result for the 2D case



(f) Result for the 1D case

Figure C.1: This figure shows a line fitting example for the 2D data points/2D triangulation case (first column) and the 2D data points/1D triangulation case (second column). The first row shows the data sets, the second row shows the triangulation, and the third row shows the SCO-RANSAC results. The severed edges are marked by a cross. In Figures (b) and (d), the outlier is shown in red to help the reader track it, but in Figure (f), the red data point is an isolated data point and represents a separate model.

The gap caused by the outlier can be solved (see Fig. C.2). We have observed that if there is a model that is assigned only one data point after the first α -expansion, then it is highly probable that it is an outlier. We refer to such data points as isolated data points, since all the incident edges on these points have already been severed. In order to avoid the gap caused by the isolated data points, before the next SCO-RANSAC iteration we remove these isolated data points from the graph and then retriangulate¹ the resulting holes. By doing this, we gain information about spatial connectivity that did not exist in the previous iteration.



(a) Remove the isolated data point and retriangulate the hole



(b) SCO-RANSAC result

Figure C.2: Figure (a) shows the result of removing the outlier (an isolated data point) and retriangulating the hole. By applying the next SCO-RANSAC iteration on the new graph, we obtain the result shown in Figure (b). This iteration makes use of the new spatial connectivity information and merges the two models. Retriangulating the hole (caused by the outlier) makes the merging feasible.

The removal of an isolated data point does not increase the energy under any circumstances. If a model is assigned a single data point then the model fits that data point perfectly (zero error); thus, removing that data point will not affect the sum of geometric errors. On the other hand, the smoothness term will always decrease, as the number of discontinues caused by having an isolated data point is equal to the number of its incident edges, while the retriangulation of the resulting hole will require a smaller number of edges/discontinuities.

¹In the case of 1D Delaunay triangulation, removing a node n and retriangulating the resulting hole is equivalent to computing the Delaunay triangulation – of the set of all nodes except n .

Bibliography

- [1] P. H. S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *IVC*, 15(8):591–605, August 1997. iii, 6
- [2] Jeffrey S. Beis and David G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, 1997. 43
- [3] M. Bertero, T. A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, 1988. 18
- [4] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, volume 1, pages 489–495, 1999. xi, 28, 43, 47, 52, 53, 54, 55, 56
- [5] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004. 37
- [6] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *International Conference on Computer Vision*, volume I, pages 377–384, 1999. 19, 22, 52
- [7] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001. 19, 25, 26
- [8] Matthew Brown and David G. Lowe. Recognising panoramas. In *ICCV*, pages 1218–1225, October 2003. 6

- [9] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *International JOURNAL of Computer Vision*, 74(1):59–73, August 2007. 6
- [10] D. Capel and A. Zisserman. Automatic mosaicing with super-resolution zoom. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 885–891, June 1998. 6
- [11] R. C. Chang and R. C. T. Lee. On the average length of delaunay triangulations. *BIT Numerical Mathematics*, 24(3):269–273, 1984. 37
- [12] Edwin K. P. Chong and Stanislaw H. Zak. *An Introduction to Optimization*. John Wiley and Sons, 2nd edition, 2001. 73, 74
- [13] M. de Berg and K. T. G. Dobrindt. On levels of detail in terrains. Technical report, CSD Utrecht University, The Netherlands, April 1995. 59
- [14] O. Faugeras and Q-T Luong. *The Geometry of Multiple Images*. MIT Press, 2004. 12, 45, 71
- [15] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981. iii, 1, 2, 22
- [16] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984. 19
- [17] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, 1988. 37
- [18] W. E. L. Grimson. *From Images to Surfaces: A Computational Study of the Human Early Visual System*. MIT Press, Cambridge, MA., 1981. 18
- [19] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003. 4, 6, 57, 70, 76

- [20] R.I. Hartley. In defense of the eight-point algorithm. *PAMI*, 19(6):580–593, June 1997. 76
- [21] Qiang He and Chee hung Henry Chu. Planar surface detection in image pairs using homographic constraints. In *ISVC*, pages 19–27, 2006. 12
- [22] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. Technical report, Cambridge, MA, USA, 1980. 18
- [23] Katsushi Ikeuchi and Berthold K. P. Horn. Numerical shape from shading and occluding boundaries. pages 245–299, 1989. 18
- [24] Y. Kanazawa and H. Kawakami. Detection of planar regions with uncalibrated stereo using distributions of feature points. In *BMVC*, volume 1, pages 247–256, September 2004. xii, 12, 57, 62, 63, 65
- [25] D. T. Lee and B. J. Schachter. Two algorithms for constructing a delaunay triangulation. *International JOURNAL of Computer and Information Sciences*, 9(3):219–242, 1980. 37
- [26] S.Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, 3rd edition, 2009. 19
- [27] G. López-Nicolás, J.J. Guerrero, O.A. Pellejero, and C. Sagüés. Computing homographies from three lines or points in an image pair. In *ICIAP*, pages 446–453, 2005. 12
- [28] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International JOURNAL of Computer Vision*, 60(2):91–110, 2004. 43
- [29] Le Lu, Xiangtian Dai, and Gregory Hager. Efficient particle filtering using ransac with application to 3d face tracking. *Image and Vision Computing*, 24(6):581–592, 2006. 6

- [30] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2009. 43
- [31] T.A. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985. 18, 19
- [32] Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, 2002. 1
- [33] C. Sagüés, A.C. Murillo, F. Escudero, and J.J. Guerrero. From lines to epipoles through planes in two views. *Pattern Recognition*, 39(3):384–393, 2006. 12
- [34] Carla M. Santos-Pereira and Ana M. Pires. Detection of outliers in multivariate data: a method based on clustering and robust estimators. In *Proceedings in Computational Statistics: 15th Symposium*, pages 291–296, 2002. 59
- [35] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6):1068–1080, 2008. 19
- [36] Ben Tordoff and David Murray. Guided sampling and consensus for motion estimation. In *ECCV*, pages 82–98. Springer-Verlag, 2002. 20
- [37] P. H. S. Torr, O. Faugeras, T. Kanade, N. Hollinghurst, J. Lasenby, M. Sabin, and A. Fitzgibbon. Geometric motion segmentation and model selection. In *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, volume 356, pages 1321–1341. The Royal Society, May 1998. 7
- [38] P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. *SPIE*, 2059:432–443, 1993. iii, 6

- [39] E. Vincent and R. Laganier. Detecting planar homographies in an image pair. In *2nd International Symposium on Image and Signal Processing and Analysis*, pages 182–187, June 2001. 2, 7, 12, 37, 38, 57, 62, 65
- [40] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005. 19
- [41] Hanzi Wang and David Suter. Mdpe: A very robust estimator for model fitting and range image segmentation. *IJCV*, 59(2):139–166, 2004. 4
- [42] Josh Wills, Sameer Agarwal, and Serge Belongie. A feature-based approach for dense segmentation and estimation of large disparity motion. *International Journal of Computer Vision (IJCV)*, 68(2):125–143, June 2006. 7, 69
- [43] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized belief propagation. In *NIPS*, volume 13, pages 689–695, 2001. 19
- [44] W. Zhang and J. Kosecka. Nonparametric estimation of multiple structures with outliers. In *WDV06*, pages 60–74, 2006. 1, 12
- [45] M. Zuliani, C. Kenney, and B. Manjunath. The multiransac algorithm and its application to detect planar homographies. In *ICIP*, volume 3, pages 153–156, 2005. 2, 7, 9, 12, 33, 37, 38, 57, 64, 65