October 2014

Energy Based Multi-Model Fitting and Matching Problems

Hossam N. Isack
The University of Western Ontario

Supervisor
Yuri Boykov
The University of Western Ontario

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

© Hossam N. Isack 2014

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Artificial Intelligence and Robotics Commons, Other Computer Sciences Commons, and the Theory and Algorithms Commons

Recommended Citation
https://ir.lib.uwo.ca/etd/2426

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact tadam@uwo.ca, wlswadmin@uwo.ca.
ENERGY BASED MULTI-MODEL FITTING AND MATCHING PROBLEMS
(Thesis format: Integrated Article)

by

Hossam Isack

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

© Hossam Isack 2014
Abstract

Feature matching and model fitting are fundamental problems in multi-view geometry. They are chicken-&-egg problems: if models are known it is easier to find matches and vice versa. Standard multi-view geometry techniques sequentially solve feature matching and model fitting as two independent problems after making fairly restrictive assumptions. For example, matching methods rely on strong discriminative power of feature descriptors, which fail for stereo images with repetitive textures or wide baseline. Also, model fitting methods assume given feature matches, which are not known a priori. Moreover, when data supports multiple models the fitting problem becomes challenging even with known matches and current methods commonly use heuristics.

One of the main contributions of this thesis is a joint formulation of fitting and matching problems. We are first to introduce an objective function combining both matching and multi-model estimation. We also propose an approximation algorithm for the corresponding NP-hard optimization problem using block-coordinate descent with respect to matching and model fitting variables. For fixed models, our method uses min-cost-max-flow based algorithm to solve a generalization of a linear assignment problem with label cost (sparsity constraint). The fixed matching case reduces to the multi-model fitting subproblem, which is interesting in its own right. In contrast to standard heuristic approaches, we introduce global objective functions for multi-model fitting using various forms of regularization (spatial smoothness and sparsity) and propose a graph-cut based optimization algorithm, PEaRL. Experimental results show that our proposed mathematical formulations and optimization algorithms improve the accuracy and robustness of model estimation over the state-of-the-art in computer vision.

Keywords: Multi-view Geometry, Fitting, Matching, Flow recycling, Label Cost.
Co-Authorship Statement

Chapter 1 is my work to summarize fitting, matching and multi-labeling problems.
Chapters 2 and 4 were a collaborative effort with my advisor, Yuri Boykov.
Chapter 3 was a close collaboration with Andrew Delong, Anton Osokin, and Yuri Boykov. My contributions were the global energy formulation and all the necessary model fitting experiments. Andrew Delong proposed the main technical idea on how to extend $\alpha$-expansion to handle label costs.
Dedication

To my loving parents, thank you for your help. I miss being with you every day.

To my lovely fiancé, thank you for your support. I promise I will always be there for you.
# Contents

Abstract .......................................................... ii

Co-Authorship Statement ......................................... iii

List of Figures ..................................................... vii

List of Tables ...................................................... xiii

1 Introduction .................................................... 1
  1.1 Multi-view Geometry Overview .............................. 1
    1.1.1 Thin Lens Model .................................. 1
    1.1.2 Pinhole Camera Model ............................. 2
    1.1.3 Stereo Vision ...................................... 5
    Plane Induced Homography ................................. 5
    Fundamental Matrix ....................................... 6
  1.2 Multi-Labeling Problem Overview ......................... 8
  1.3 Thesis Overview ........................................... 12
    1.3.1 Basic Problem Formulation ......................... 12
    1.3.2 Summary of Contributions .......................... 13
    1.3.3 Thesis Structure .................................. 16

2 Energy-based Geometric Multi-Model Fitting ............ 19
  2.1 Introduction ................................................ 20
    2.1.1 Towards Energy Optimization ....................... 21
    2.1.2 Related Work ...................................... 26
  2.2 Our Approach (PEARL) ..................................... 28
    2.2.1 Proposing Initial Labels ............................ 28
    2.2.2 Energy Formulation ................................ 29
    2.2.3 Outliers ........................................... 30
    2.2.4 Expand and Re-estimate Labels ...................... 31
    2.2.5 Algorithm and Its Properties ....................... 32
  2.3 Experimental Results ...................................... 39
    2.3.1 Estimating Multiple Affine Models ................. 40
    2.3.2 Estimating Multiple Homographies .................. 43
    2.3.3 Motion Segmentation ................................ 48
    Using two-views .......................................... 48
Using multiple views/frames ........................................... 53
2.4 Conclusions ......................................................... 55

3 Fast Approximate Energy Minimization with Label Costs 60
3.1 Some Useful Regularization Energies ............................ 60
3.2 Fast Algorithms to Minimize (⋆) .................................. 65
  3.2.1 Expansion moves with label costs ............................ 65
  3.2.2 Optimality guarantees ......................................... 67
  3.2.3 Easy case: only per-label costs ................................ 69
  3.2.4 Energy (⋆) as an information criterion ....................... 70
3.3 Working With a Continuum of Labels ............................ 70
3.4 Applications and Experimental Setup ............................ 73
  3.4.1 Geometric multi-model fitting ............................... 75
  3.4.2 Image segmentation by MDL criterion ....................... 77
3.5 Discussion .......................................................... 77

4 Energy based multi-model fitting & matching for 3D reconstruction 80
4.1 Introduction .......................................................... 80
4.2 Fit-&-Match Energy .................................................. 82
4.3 Overview of Our Approach ......................................... 83
4.4 Optimization .......................................................... 84
  4.4.1 Solving GAP .................................................... 85
    Reducing GAP to LAP ............................................... 85
    LAP as MCMF (overview) .......................................... 85
    Solving MCMF (overview) ......................................... 86
    Solving a Series of GAPs ......................................... 86
  4.4.2 Local Search-GAP (LS-GAP) .................................. 86
4.5 Evaluation ............................................................ 87
4.6 Conclusions .......................................................... 93

5 Conclusion ............................................................. 94

Curriculum Vitae .......................................................... 95
List of Figures

1.1 (a) shows a thin lens model and (b) shows its basic functional properties. 2

1.2 (a) shows a pinhole camera model and (b) shows the pinhole mathematical model with an “imaginary” image plane in front of the center of projection, also the camera coordinate system $C$ is located at the center of projection. (c) shows the image plane coordinate system $I$ relative to the digital image coordinate system $D$. (d) shows a rigid transformation $[R|T]$ tying the camera coordinate system $C$ and the world coordinate system $W$. 3

1.3 (a) and (b) show plane projection in single and stereo camera settings, respectively. 6

1.4 (a) Stereo images showing epipoles and epipolar line corresponding to $x$. Figure (b) shows how knowing matches in a stereo setting allows us to recover 3D information of scene points. 8

1.5 (a) is the image to be segmented. (b) and (c) show segmentation results where the foreground is red shaded and the background is blue shaded. Labeling the white part of the polar bear using only color cue is not sufficient as it is equally likely to be foreground or background. When labeling variations is discouraged as in (c) labeling the white part of the polar bear as a foreground becomes a more favorable solution. 9

1.6 (a) graph nodes. Figures (b) and (c) show the edges corresponding to the unary and pairwise potentials of equation (1.20), respectively. The constructed graph $G$ is shown in (d). Figure (e) shows a possible $s \backslash t$-cut and (f) shows its corresponding pixel labeling. In (e) the severed edges by the $s \backslash t$-cut are shown in gray color. 10

1.7 (a) shows a pair of images with parts (green and blue) that undergo different translations. (b) shows feature sets $\mathcal{F}_i$ and $\mathcal{F}_r$, and a possible match. (c) shows translations $\theta_i$ and $\theta_j$ that the green and blue parts undergo. (d) shows an ideal labeling: green’s part features (inside green cloud) are labeled $i$ and blue’s part features (inside blue cloud) are labeled $j$. 14
2.1 Blue dots are data points supporting 8 lines. In multi-model cases, detect-
ing models by maximizing the number of inliers may work for low levels
of noise (a). Higher noise levels require larger thresholds to detect inliers
(b), but then some random model (red) may have far more inliers than the
true model (green). The integers show the number of model inliers for the
selected thresholds. Simplistic greedy selection of models with the largest
number of inliers would fail in (b). This example illustrates a general prob-
lem that affects many multi-model fitting approaches that greedily selects
one model at a time while ignoring the overall solution. It also motivates
our global energy approach that considers the quality of the whole solution.

2.2 Motivating spatial regularization in geometric model fitting. In many vi-
sion problems combining geometric errors and spatial coherence terms in
energy (2.3) can be justified *generatively* because clusters of inliers are
generated by regular objects (a). Moreover, spatial regularization can also
be justified *discriminatively*. Plot (b) shows an average deviation from
the ground truth for optimal models obtained in 100 randomly generated
line-fitting tests as in Fig. 2.4(d). Each point in this plot corresponds to
some fixed smoothness parameter $\lambda$ in (2.3). Clearly, the spatial coherence
term significantly reduces estimation errors for some $\lambda > 0$.

2.3 Examples of inlier classification from thresholding. If data is known to
support one model (a) then thresholding identifies inliers (blue) for any
model without ambiguity. In case of 3 models (b), simple thresholding
may not disambiguate inliers (red) between the models.

2.4 Illustration of PEARL’s iterations. (a) proposals generated by random
sampling, (b-d) re-estimation of models and their inliers after several iter-
ations of *expand* and *re-estimate* steps for energy (2.3) or (2.7). Note that
the algorithm may converge to good models even from a small set of rough
guesses. In this example we did not use an *outlier label*, see Section 2.2.3,
and an optimal set of lines in (d) “explains” all data points.

2.5 We use Delaunay triangulations of data points. K-nearest points or other
techniques can be used as well, particularly for higher dimensional data.
We did not observe much difference in practical performance.

2.6 PEARL vs RANSAC for synthetic data sets with 80% uniformly dis-
tributed outliers and 20% noisy inliers supporting one line model. In
each test both methods used the same initial set of randomly sampled
model proposals. X-axis is the number of initially proposed models. Y-
axis shows estimation errors w.r.t. the parameters of the ground truth
line model. The errors are averaged over 400 randomly generated tests.
RANSAC basically selects the best model from the initial set of propos-
als. By iteratively minimizing global energy (2.3) PEARL can converge
to a much better model than those in the initial set. In contrast, stan-
dard RANSAC strongly relies on a larger number of initial samples to find
accurate estimates.
2.7 Comparing the results for fitting lines to noisy data points. The data points were perturbed with a low level of Gaussian noise (σ = 0.01) and 200 random outliers were added. Outliers represent 25% of the data.

2.8 Comparing the results for fitting lines to noisy data points. The data points were perturbed with medium Gaussian noise (σ = 0.02) and included 300 random outliers. Outliers represent 35% of the data.

2.9 Comparing the results for fitting lines to noisy data points. The data points were perturbed with high Gaussian noise (σ = 0.025) and 500 random outliers were added. Outliers represent 45% of the data.

2.10 Intersecting lines example. Optimization of energy (2.3) may leave spatially isolated groups of inliers assigned to 2 models even if their parameters are infinitesimally close (b). Per-label costs in energy (2.7) solves this problem (c) but requires either an additional merging operation or an extension of α-expansion—introduced in Chapter 3. This example motivates the use of the EM-style soft assignments of labels to points. However, in computer vision as will be shown in Sec. 2.3 we get occlusions (not intersections), which better motivates our “hard” assignments of labels.

2.11 Fitting lines with different noise levels. The inliers in (a) were generated with different levels of Gaussian noise. 40% of the data are outliers. In (b) PEARL estimated labels combining geometric model parameters with unknown noise variances using error measure (2.8). Previous multi-model fitting methods use fixed thresholds to identify inliers, which would not work in this case.

2.12 Geometric fitting errors ||p − A|| for rectified stereo. Assuming matched points p = {p_l, p_r} are on the corresponding epipolar (scan) lines \( \Delta_y = y_l − y_r = 0 \), the left transfer error is a horizontal shift \( \Delta_x \) between \( p_r \) and \( A p_l \) as shown in (a). The standard re-projection error (b) is obtained by treating points \( p_l \) and \( p_r \) as noisy observations of some hidden “true” pair of matched points \( \tilde{p} = {\tilde{p}_l, \tilde{p}_r} \) where \( \tilde{p}_r = A \tilde{p}_l \) and \( \tilde{p} \) minimizes the observation noise min \( |\tilde{p}_l − p_l|^2 + |\tilde{p}_r − p_r|^2 \).

2.13 Results for ”Clorox” stereo pair [2]. (a) Dense pixel segmentation by BT [2] uses photo-consistency. (b-c) Sparse inlier classification by PEARL using geometric fit measures (2.9) and (2.10), respectively.

2.14 Comparison of results by BT [2] and PEARL. Lines are computed for all pairs of intersecting planes (affine models).

2.15 Typical results for affine model fitting in Sec. 2.3.1 via sequential RANSAC (a) and J-linkage [26]. We used 5000 uniform samples for RANSAC. Different threshold values did not give much improvement. Similarly, different tunings of J-linkage generated various artifacts. These problems might be explained by the large level of noise in the data, as in Fig. 2.9.

2.16 Multi-homography fitting, images from VGG (Oxford) Merton College I.

2.17 Multi-homography fitting for stereo images from VGG (Oxford) Merton College III. Both results were obtained using standard SIFT features.
2.18 Multi-homography fitting for “stairs” stereo image from VGG (Oxford). The results in (b) and (c) represent our best efforts in tuning RANSAC and J-linkage thresholds. For example, two “walls” start to grab false positive matches for larger threshold values, while for smaller thresholds the “stairs” become even more over-segmented compared to what is shown here. .......................... 46

2.19 In this Raglan Castle example from VGG (Oxford) we used the same color more than once to represent different planes (a). Only spatially connected planes are shown in different colors. An image of the same scene from a different view (b) confirms that the vertical walls of the building corresponds to different planes. .......................... 47

2.20 Representative PEARL's results for motion sequences from Rene Vidal's data set [31]. Our motion estimation using mixtures of fundamental matrices (Sec. 2.3.3) uses only two frames (the first and the last). .......................... 50

2.21 Optimal solution for energy (2.3). In contrast to results for energy (2.7) in Fig.2.20 (e)-(h), it fails to generate one background motion from yellow, green, and red clusters. These clusters correspond to infinitesimally close motions, but they are spatially isolated. The third term in (2.7) addresses this issue. .......................... 51

2.22 Histogram of misclassification errors for PEARL with energy (2.13) on 26 Checkerboard examples [31] with 3 distinct motions. This histogram reveals two “modes”. The mode with smaller misclassification errors mainly contains examples where PEARL produced exactly 3 models. Such examples are marked in blue. Examples where PEARL produced a different number of models are marked in red. Such examples formed the “gross errors” mode. The percentage of misclassified points may not be a proper measure for comparing a method that automatically computes the number of models against the methods that a priori know the correct number. We separately report PEARL's error statistics for 15 (blue) examples with the correct number of obtained models, see ER1 in Table 2.2. .......................... 51

2.23 Scatter plot of different solutions obtained by PEARL on one of the examples in [31] using different sets of 5000 (uniformly) sampled fundamental matrices. This number of samples is statistically insufficient for the motion estimation examples in Sec. 2.3.3 (see text) and the quality of optimization may suffer. The plot also illustrates positive correlation between the values of our energy (2.13) and the misclassification errors. .......................... 52

2.24 Representative PEARL’s results using energy (2.15) for motion sequences from Vidal’s data set [31]. Unlike the fundamental matrix fitting method in Sec. 2.3.3, this formulation uses all frames in the sequence. .......................... 54

2.25 Multi-motion estimation accuracy in one of the checkerboard sequences from Sec. 2.3.3. Misclassification errors are shown for optimal solutions corresponding to different values of $\lambda$ and $\beta$ in energy (2.15). .......................... 55
3.1 Motion segmentation on the 1RT2RCR sequence [25]. Energy (3.1) finds 3 dominant motions (b) but labels many points incorrectly. Energy (3.2) gives coherent segmentations (c) but finds redundant motions. Our energy (⋆) combines the best of both (d).

3.2 Planar homography detection on VGG (Oxford) Merton College 1 image (right view). Energy (3.1) finds reasonable parameters for only the strongest 3 models shown in (b), and still assigns a few incorrect labels. Energy (3.2) finds reasonable clusters (c) but fits 9 models, some of which are redundant (nearly co-planar). Our energy (⋆) finds both good parameters and labels (d) for 7 models.

3.3 Unsupervised segmentation using histogram models. Energy (3.1) clusters in colour space, so segments (b) are incoherent. Energy (3.2) clusters over pixels and must either over-segment or over-smooth (c), just as in [26]. Our energy (⋆) balances these criteria (d) and corresponds to Zhu & Yuille [27] for segmentation.

3.4 LEFT: Graph construction that encodes $h - h x_1 x_2 \cdots x_k$ when we define $x_p = 1 \leftrightarrow p \in T$ where $T$ is the sink side of the cut. RIGHT: In a minimum $s$-$t$ cut, the subgraph contributes cost either 0 (all $x_p = 1$) or $h$ (otherwise). A cost greater than $h$ (e.g. *) cannot be minimal because setting $y = 0$ cuts only one arc.

3.5 Re-estimation helps to align models over time. Above shows 900 raw data points with 50% generated from 5 line intervals. Random sampling proposes a list of candidate lines (we show 20 out of 100). The 1st segmentation and re-estimation corresponds to Li [18], but only the yellow line and gray line were correctly aligned. The decreasing energies in Figure 3.8 correspond to better alignments like the subsequent iterations above. If a model loses enough inliers during this process, it is dropped due to label cost (dark blue line).

3.6 We can also fit line intervals to the raw data in Figure 3.5. The three results above were each computed from a different set $L$ of random initial proposals. See Section 3.4.1 for details.

3.7 For multi-model fitting, each label can represent a specific model from any family (Gaussians, lines, circles...). Above shows circle-fitting by minimizing geometric error of points.

3.8 Energy (⋆) over time for a line-fitting example (1000 points, 40% outliers, 6 ground truth models). Only label cost regularization was used. Re-estimation reduces energy faster and from fewer samples. The first point (●) in each series is taken after exactly one segmentation/re-estimation, and thus suggests the quality of Li [18] using a greedy algorithm instead of LP relaxation.

3.9 Unsupervised segmentation by clustering simultaneously over pixels and colour space using Gaussian mixtures (colour images) and non-parametric histograms (greyscale images). Notice we find coarser clustering on baseball than Zabih & Kolmogorov [26] without over-smoothing. For segmentation, our energy is closer to Zhu & Yuille [27] but our algorithm is more powerful than region-competition.
4.1 Best seen in Color, Fig. (a) shows the effect of EFM iterations on energy (4.1). EFM converged on the average after 5 iterations, and each iteration on the average took 1 min. Figure (b) shows multiple histograms of the final energies for different sizes of initial set of proposals—blue indicates a large set of proposals. The larger the set of initial proposals $L$ the more likely that EFM will converge to a low energy.

4.2 Best seen in Color, Fig. shows ROC curve for standard SIFT matches by varying SBR threshold, and the scatter plot represents EFM results for different sizes of initial set or proposals. As can be seen, the lower the final energy (blue) the better the matching.

4.3 Best seen in Color, Fig. (a) shows EF labeling result (average TPR=0.51 and FPR=1.6E-05) and (b) shows EFM labeling result (average TPR=0.98 and FPR=9.1E-06). Features assigned to the same model/label are drawn in the same color and unmatched features are shown as white x. Figures (c-f) show the enlargement of segments 1 and 2 in (a) and (b). Figures (g-h) show the matching, between two small regions in the stereo images, of the EF and EFM results, respectively.

4.4 Best in Color, the first column shows one of the stereo images for each example, second and third columns show the EF and EFM labelling results superimposed on the images shown in the first column, respectively. On average of 50 runs, EFM found 0.75, 10.53, 3.31, 0.44, and 0.68 times more inliers than EF. EFM and EF+GM results where comparable, EFM found approximately between 0.05 and 0.08 times more matches.
List of Tables

2.1 Geometric errors for lines in Fig. 2.14 where affine models intersect. Errors are computed as the sum of distances between the ground truth line segment corners and the computed lines. The errors for sequential RANSAC and J-linkage, see Fig. 2.15, were significantly larger than those above. ................................. 43

2.2 Misclassification errors for checkerboard motion estimation sequences in [31]. The results for REF, GPCA [32], LSA [36], and RANSAC are copied from Table 4 in [31]. These methods use all N-frames in each video sequence. In contrast, our results for PEARL with energy (2.13) use only 2 frames, the first and the last in the sequence. Other methods also “know” that the exact number of models is 3. In contrast, PEARL may obtain a different number of models. Figure 2.22 illustrates how this affects PEARL’s classification errors. For a more balanced comparison we show 2 additional statistics. First, column ER1 evaluates 15 out of 26 examples where PEARL obtained exactly 3 models. Second, similarly to REF [31], column ER2 evaluates the “reference” solutions obtained by PEARL from the ground truth models in all 26 examples. .................. 49

2.3 Misclassification errors for checkerboard motion estimation sequences in [31]. The results for REF, GPCA [32], LSA [36], and RANSAC are copied from Table 4 in [31]. All methods in this table use all frames in each video sequence. PEARL may obtain a different number of models while the other methods “know” that the exact number of models is 3. For a more balanced comparison we show 2 additional statistics. First, column ER1 evaluates 12 out of 26 examples where PEARL obtained exactly 3 models. Second, similarly to REF in [31], column ER2 evaluates the “reference” solutions obtained by PEARL from the ground truth models in all 26 examples. .................. 56

4.1 Graphite VGG Oxford dataset, single model and increasing baseline. The table shows the averages of GQ and ROC attributes, over 50 runs, for EFM, EF, and EF+GM model estimates. .................. 90
Chapter 1

Introduction

Sections 1.1 and 1.2 are intended for readers who are not familiar with multi-view geometry, and common multi-labeling problems in computer vision and their optimization frameworks. In Section 1.1, we will overview some basic multi-view geometry principles [10, 7] e.g. camera models and stereo vision. Then, in Section 1.2, we will discuss a common multi-labeling problem and give a brief overview of the α-expansion algorithm [4] which is widely used in computer vision for multi-labeling problems. Finally, in Section 1.3 we overview the problems we are trying to solve and summarize our contributions.

1.1 Multi-view Geometry Overview

We will start by studying some basic camera models, i.e. thin lens and pinhole camera. Then we will discuss plane induced Homographies and the fundamental matrix which could be used to detect planes and rigid motions in a scene, respectively.

1.1.1 Thin Lens Model

For simplicity, the following discussion is limited to converging thin lens. In a nutshell, a thin lens has a negligible thickness, i.e. the distance between the two surfaces of the lens, compared to the radii of curvature of the lens’ surfaces. Let us denote the optical center of the lens by \( C \) see Fig. 1.1(a). Also, let the optical axis be the line perpendicular to the lens and passing through \( C \). The focal length \( f \) is the main parameter of any lens. Knowing the focal length of a converging thin lens allows us to identify two focal points on each side of the lens, and each focal point is along the optical axis and at a distance \( f \) from \( C \). The functional properties of a thin lens, as shown in Fig. 1.1(b), are:

- The direction of any ray entering the lens through the optical center will not be affected.
- Any ray entering the lens parallel to the lens’s optical axis on one side will be refracted on the other side to intersect the optical axis at a distance \( f \) from the optical center.
- Any ray that enters the lens after passing through the focal point from one side will come out parallel to the optical axis on the other side.
Chapter 1. Introduction

(a) Thin lens  (b) Functional properties

Figure 1.1: (a) shows a thin lens model and (b) shows its basic functional properties.

Another parameter is the aperture opening/hole of diameter $D$ which restricts the range of rays passing through or reaching the lens, as the aperture opening could be located on either side of the lens, see Fig. 1.1(a). An ideal thin lens model has an infinitesimal aperture which in theory only allows rays that are going through the optical center resulting in a purely refractive lens [14]. This idealization is also known as pinhole camera model.

1.1.2 Pinhole Camera Model

The pinhole camera model is one of the simplest and oldest camera models, Fig. 1.2(a). Other camera models e.g. affine camera models, orthographic projection model etc. are discussed in [10]. A full and comprehensive discussion on camera models is beyond the scope of this work. In this work we discuss the pinhole model since it is considered a good geometric approximation of the ideal thin lens—with infinitesimally small aperture.

As can be seen in Fig. 1.2(a) the formed image is always flipped horizontally and vertically on the image plane. In our pinhole mathematical model we adopt the frontal image plane approach where an imaginary plane is placed in front of the center of projection. To describe the mathematical model we first need to define two coordinate systems: camera coordinate system $\mathbf{C}$ with the basis $\{e^C_x, e^C_y, e^C_z\}$ and the image plane coordinate system $\mathbf{I}$ with the basis $\{e^I_x, e^I_y\}$, see Fig. 1.2(b). Also, the image plane is placed at a distance $f$ from $\mathbf{C}$ and with a normal $e^C_z$. Later on we will define other coordinate systems and, to avoid any ambiguity, the superscript of a vector in our notation will denote which coordinate system the vector is defined with respect to, e.g. $e^C_x$ is a vector defined with respect to $\mathbf{C}$.

Let $V^C = [a, b, c]^T$ be a point on the imaged object and $V^I = [u, v]^T$ be its projection on the image plane. Knowing that the image plane is at $f$ with normal $e^C_z$, by projecting $V^C$ on $\{e^C_x, e^C_y\}$ and $\{e^C_y, e^C_z\}$, and using triangle similarities we can deduce that

\begin{align}
    u &= f \frac{a}{c} \\
    v &= f \frac{b}{c},
\end{align}

(1.1)

(1.2)
1.1. Multi-view Geometry Overview

Figure 1.2: (a) shows a pinhole camera model and (b) shows the pinhole mathematical model with an “imaginary” image plane in front of the center of projection, also the camera coordinate system C is located at the center of projection. (c) shows the image plane coordinate system I relative to the digital image coordinate system D. (d) shows a rigid transformation $[R|T]$ tying the camera coordinate system C and the world coordinate system W.
Chapter 1. Introduction

To avoid the nonlinearity in relating $V^I$ and $V^C$ in equations (1.1) and (1.2), we will use homogeneous coordinates to represent $V^I$ instead of Euclidean coordinates. Let $X = [x_1, x_2, \ldots, x_n]^T$ be a point in $\mathbb{R}^n$. To represent this point in homogeneous coordinates we add a coordinate and set it to 1, $X = [x_1, x_2, \ldots, x_n, 1]^T$. In our notation $X$ is the homogeneous representation of $X$. In Projective coordinate systems where homogeneous coordinates are used scaling is not important 

$$[x_1, x_2, \ldots, x_n, 1]^T \simeq [\lambda x_1, \lambda x_2, \ldots, \lambda x_n, \lambda]^T$$

where $A \simeq B$ means that there exists a scale $c \neq 0$ such that $A = cB$. By using the homogeneous coordinates of $V^I$, we can linearly relate $V^I$ to $V^C$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}.
\tag{1.3}$$

The digital image, is a discretized version of the image plane, see Fig. 1.2(c). In the digital image the origin is usually not in the middle of the image and the coordinates are in pixels not metric units like the image plane coordinates. Let the coordinate system $D$ with the basis $\{e^D_x, e^D_y\}$ be the digital image coordinate system and $V^D = [x, y]^T$ be the transformed $V^I = [u, v]$ from the $I$ coordinate system, see Fig. 1.2(c). Also, let $m_u$ and $m_v$ be the pixels/unit resolution of the digital image and $[t_u, t_v]$ be the translation from $I$ to $D$. Now we can relate $V^D$ to $V^C$ as follows

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \simeq \begin{bmatrix} m_u & 0 & 0 \\ 0 & m_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_u \\ 0 & 1 & t_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}.
\tag{1.4}$$

In real life we never know the scene points coordinates with respect to the camera coordinate system $C$. However, it is possible to introduce a new coordinate system, namely world coordinate system $W$ with the basis $\{e^W_x, e^W_y, e^W_z\}$, to define the scene points. The world coordinate system could be chosen arbitrarily, see Fig. 1.2(d). Also, let $[R|T]$ be the 3D transformation that transforms $W$ to $C$ where $R$ and $T$ are the 3D rotation and translation, respectively. Let $V^W = [X, Y, Z]^T$ be the scene point defined with respect to $W$ such that

$$V^I = [R|T]V^W.
\tag{1.5}$$

Using (1.5) and (1.4) we can relate a world point $V^W$ to its pixel location $V^D$ on the digital image as follows
1.1. Multi-view Geometry Overview

\[
\begin{bmatrix}
x \\
y \\
1 
\end{bmatrix} \approx 
\begin{bmatrix}
m_u f & 0 & m_u t_u \\ 0 & m_v f & m_v t_v \\ 0 & 0 & 1 
\end{bmatrix} 
\begin{bmatrix}
X \\
Y \\
Z 
\end{bmatrix}
\] (1.6)

\[
V^D \approx K[R|T]V^W
\] (1.7)

\[
V^D \approx PV^W.
\] (1.8)

The upper triangular $3 \times 3$ matrix $K$ in (1.7) is commonly referred to as the camera matrix and it contains all the intrinsic camera parameters, e.g. focal length and resolution. The $R$ and $T$ are extrinsic parameters, as they define the camera position relative to a world coordinate system of our choosing. Finally, the $3 \times 4$ matrix $P$ in (1.8) denotes the projection matrix of a pinhole camera.

1.1.3 Stereo Vision

Capturing a scene from one viewpoint, i.e. a camera, is not enough to recover depth information to reconstruct the 3D scene—an image point could be the image of any 3D point along the ray from the camera center of projection through that point. In stereo vision, two cameras are used to capture a scene from two different viewpoints. The stereo setting imitates human vision. Using a pair of stereo images one could recover depth information, i.e. reconstruct the scene. Also, planes in the scene could be detected by computing their plane induced homographies. There are different kinds of 3D reconstructions, e.g. projective, euclidian and metric reconstruction, see [10] for more details. In this work we will not discuss 3D reconstruction methods but rather an essential piece of information used in 3D reconstruction, i.e. the fundamental matrix.

Plane Induced Homography

A set of planar points and their projections onto a digital image could be related by a projective transformation, namely homography. In order to simply our derivation of the homography transformation, we will choose the world coordinate system $W$ such that its $\{e^x_w, e^y_w\}$ basis span the 2D plane, see Fig. 1.3(a). Let $V^W = [X,Y,Z]^T$ be a world point that belongs to the plane and $V^D = [x,y]^T$ be its projection on to the digital image. Note that any point that belongs to the plane will have $Z = 0$ by construction. From equation 1.6 we can deduce the following

\[
\begin{bmatrix}
x \\
y \\
1 
\end{bmatrix} \approx 
\begin{bmatrix}
m_u f & 0 & m_u t_u \\ 0 & m_v f & m_v t_v \\ 0 & 0 & 1 
\end{bmatrix} 
\begin{bmatrix}
r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 
\end{bmatrix} 
\begin{bmatrix}
X \\
Y \\
0 \\
1 
\end{bmatrix}
\] (1.9)

\[
\begin{bmatrix}
x \\
y \\
1 
\end{bmatrix} \approx 
\begin{bmatrix}
m_u f & 0 & m_u t_u \\ 0 & m_v f & m_v t_v \\ 0 & 0 & 1 
\end{bmatrix} 
\begin{bmatrix}
r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 
\end{bmatrix} 
\begin{bmatrix}
X \\
Y \\
1 
\end{bmatrix}.
\] (1.10)
A homography in equation (1.10) is a $3 \times 3$ matrix. It should be noted that a general homography has 8 degrees of freedom not 9 since we could always set one of the 9 parameters to 1 to enforce equality in (1.10).

In the stereo setting shown in Fig. 1.3(b), let the two cameras’ center of projections be denoted by $C_1$ and $C_2$, respectively. Also, let the world plane be projected onto the two images by the homographies $H_1$ and $H_2$. The projective transformation $H$ that relates the plane projections on these two images is also a homography

$$H = H_2H_1^{-1}.$$ 

In this example, $H$ will transform the projected image of the plane from the first image to the second one.

![Diagram](image)

(a) Single camera and projected plane  
(b) Stereo cameras and projected plane

Figure 1.3: (a) and (b) show plane projection in single and stereo camera settings, respectively.

**Fundamental Matrix**

Before we can define and derive the fundamental matrix we will cover some basic stereo vision concepts. As we mentioned earlier, depth information cannot be recovered using only one camera. Using only one camera, if $x$ is the projection of a world point $X$ then we can deduce that $X$ lies on the ray from the camera center of projection to $x$, see Fig. 1.4(a) consider only the left camera. Now let us consider a stereo setting where the cameras’ relative orientation is known, as in Fig. 1.4(a). In that case we can easily deduce that the projection of $X$ onto the second camera can only lie on the line of intersection between the camera’s image plane and the $xCC’$ plane, i.e the epipolar plane. We will refer to that line as the epipolar line corresponding to $x$, see Fig. 1.4(a). Also, we will refer to the points of intersection between the images’ planes and $CC’$ by epipoles $e$ and $e’$, see Fig. 1.4(a), and $CC''$ by the baseline. If we know that $x$ and $x'$ are the projections of the same world point onto two images, we can infer the world point $X$ by finding the intersection point of the following viewing rays $Cx$ and $C’x’$, see Fig. 1.4(b). The problem of inferring the world point is referred to as the triangulation problem in computer vision. We initially assumed that we know the relative position of the cameras but in real life this piece of information is rarely known. The fundamental matrix embeds the relative
position of the two cameras and their camera matrices, and this is why it is a vital part for 3D reconstruction.

The fundamental matrix $F$ is a transformation. It transforms a point, e.g. $x$, from the left image to its corresponding epipolar line, i.e. line containing $e'$ and $x'$, in the right image, see Fig. 1.4(b),

$$Fx \simeq e' \times x'. \quad (1.11)$$

Recall that $x$ is the homogeneous coordinate representation of $x$. In our derivation of the fundamental matrix, we will denote the projection matrices of the left and right images by $P$ and $P'$, respectively. The epipole $e'$ is the projection of $C$ onto the image plane of $C'$

$$e' \simeq P'C. \quad (1.12)$$

Furthermore, let $x$ be the projection of $X$ on $C$, i.e. $x \simeq PX$. Let $P^+$ be the pseudo inverse of $P$ and since $P$ is a rectangular matrix with linearly independent rows then $PP^+ = I$. The world point $P^+x$ belongs to the line $CX$, i.e. its projection on $C$ is $x$, since $x \simeq PP^+x$.

In addition, let $\hat{x}'$ be the projection of $P^+x$ on $C'$

$$\hat{x}' \simeq P'P^+x. \quad (1.13)$$

The point $\hat{x}'$ lies on the line $e'x'$ because the line $CX$ is projected on to $C'$ over the line $e'x'$. Therefore,

$$Fx \simeq e' \times \hat{x}'. \quad (1.14)$$

In equation (1.14) we could substitute $e'$ and $\hat{x}'$ from (1.12) and (1.13), respectively

$$Fx \simeq (P'C) \times (P'P^+x). \quad (1.15)$$

Also, $C$ and $C'$ are related by unknown $[R|T]$ by construction. Now let us choose the world coordinate system to be $C$, i.e. $C = 0$ where $0$ is the zero vector. Then we can deduce the following

$$P = K[I|0], \quad (1.16)$$

$$P' = K'[R|T], \quad (1.17)$$

$$P^+ = \left[ \begin{array}{c} K^{-1} \\ 0^T \end{array} \right]. \quad (1.18)$$

Using (1.15) to (1.18), we can represent $F$ in terms of the camera matrices and their relative position as follows

$$Fx \simeq (P'C) \times (P'P^+x) \simeq (K'T) \times (K'RK^{-1}x) \simeq [K'T] \times K'RK^{-1}x \quad \text{(Fundamental matrix)} \quad (1.19)$$
Chapter 1. Introduction

Figure 1.4: (a) Stereo images showing epipoles and epipolar line corresponding to \( x \). Figure (b) shows how knowing matches in a stereo setting allows us to recover 3D information of scene points.

where \([v]_\times\) denotes the skew symmetric matrix representation\(^1\) of vector \( v \), e.g. let \( v = [a, b, c]^T \) then

\[
[v]_\times = \begin{bmatrix}
0 & -c & b \\
-c & 0 & -a \\
-b & a & 0
\end{bmatrix}.
\]

The fundamental matrix could be used to track rigid motions in a scene [12]. Our derivation of the fundamental matrix came from a stereo setting where we used two cameras and the scene/object was static. Another way to arrive to the same conclusion is to use one none moving camera and take two pictures of a rigidly moving object/scene. In that case, the \( R \) and \( T \) in (1.19) will correspond to the rigid-body’s motion. Furthermore, if there is more than one object undergoing different rigid motions each of them will have it is own fundamental matrix. Thus, we can detect and identify multiple rigid motions in a scene by estimating multiple fundamental matrices.

1.2 Multi-Labeling Problem Overview

In computer vision many problems require assigning a label to every pixel, feature or substructure of an image. These labels could be color models, disparities, objects etc. Also, an intuitive assumption is that nearby pixels in a 2D image are more likely to be the projections of the same 3D object compared to distant ones. Let \( \mathcal{L} \) denote the set of labels. We will start by discussing the simple case where \(|\mathcal{L}| = 2\) and show how to find the optimal labeling in polynomial time. Then, we will show how this idea was extended in [4], \( \alpha \)-expansion, to find an approximate solution for the case where \(|\mathcal{L}| \geq 3\).

\(^1\)Note that \( v \times x = [v]_\times x \).
1.2. Multi-Labeling Problem Overview

(a) Source image  
(b) Greedy segmentation  
(c) Desired segmentation respects object boundaries

Figure 1.5: (a) is the image to be segmented. (b) and (c) show segmentation results where the foreground is red shaded and the background is blue shaded. Labeling the white part of the polar bear using only color cue is not sufficient as it is equally likely to be foreground or background. When labeling variations is discouraged as in (c) labeling the white part of the polar bear as a foreground becomes a more favorable solution.

Case \(|\mathcal{L}| = 2\): One common problem in computer vision is the foreground/background segmentation [15, 3] where it is required to estimate the foreground and background color models of an image and to label (or cluster) each pixel either as a foreground or background. For simplicity, we will assume that the color models are given and we only need to label the pixels. One could use unary-potential decisions that would label a pixel foreground if it is more likely to belong to the foreground color model compared to the background one, and label it background otherwise. However, that does not usually yield the desired results that would respect object boundaries as in Fig. 1.5(b). The reason for this is that in the greedy approach the labeling/clustering decision of each pixel is taken independently of the other pixels. A common prior in computer vision, i.e. spatial smoothness, is that the label variation between nearby pixels should be discouraged everywhere except at sharp discontinuities that may exist, e.g. object boundaries, see Fig. 1.5(c).

The foreground/background segmentation problem could be formulated as an energy functional that discourages discontinuities but first we will introduce some notation. Let \(\mathcal{P}\) be the set of image pixels to be labeled. Also, let \(\mathcal{N}\) be some pixel neighborhood\(^2\) edges where for every neighboring pixels \(p\) and \(q\) in \(\mathcal{P}\) there is an edge \((p, q) \in \mathcal{N}\). We also define the labeling variables as follows: \(f_p\) is the labeling variable of pixel \(p\) in \(\mathcal{P}\) and \(f = \{f_p \mid \forall p \in \mathcal{P}\}\). The foreground/background segmentation energy is

\[
E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \lambda \sum_{(p,q) \in \mathcal{N}} [f_p \neq f_q] \tag{1.20}
\]

\(^2\)Two commonly used neighborhoods in computer vision for 2D images are the 4-neighborhood and 8-neighborhood where a pixel is considered in the neighborhood of its 4 (or 8) nearest pixels.
where \( f_p \) could be either 0 for foreground or 1 for background, \( D_p(f_p) \) is the penalty\(^3\) of labeling pixel \( p \) as \( f_p \), and \([\ ]\) are the Iverson bracket—if the condition inside the brackets is true the Iverson brackets return 1 and 0 otherwise. The data term is a unary potential and minimizing it (without the smoothness term) could be done using simple thresholding, see Fig. 1.5(b). The smoothness term is a pairwise potential that discourages labeling variation between neighboring pixels. The smoothness parameter \( \lambda \) is a normalization parameter; when it is \( \infty \) all the pixels will be assigned to the same label, and when it is 0 the pixel labeling will be done independently for each pixel. For the right smoothness parameter we could achieve the desired segmentation as shown in Fig. 1.5(c).

When we only have two labels, the optimal labeling could be found in polynomial time. It is possible to formulate the minimization problem (1.20) as a min-cut problem. Furthermore, the min-cut problem is the dual of the max-flow problem which could

\(\text{\footnote{For foreground/background segmentation it is the } -\log \text{ of the likelihood of } p \text{ belonging to label } f_p.}\)
be solved in polynomial time. We will only cover how to formulate (1.20) as a min-cut problem. However, the discussion on the duality between min-cut and max-flow problems is beyond the scope of this work, see [18] for a detailed discussion.

To formulate the minimization of (1.20) as a min-cut problem, we will construct a graph $G$ with two special nodes $s$ and $t$ where there is a one to one correspondence between an $s\backslash t$-cut \footnote{An $s\backslash t$-cut will partition a graph $G$ into two disjoint sets such that $s$ and $t$ belong to different sets.} and pixel labeling. For example, in foreground/background segmentation if a node is connected to $s$ in an $s\backslash t$-cut then it belongs to the foreground otherwise it belongs to the background. In a nutshell, the main idea is to construct a graph where every valid $s\backslash t$-cut would corresponds to a unique valid pixel labeling, and that the $s\backslash t$-cut cost would be equal to its corresponding pixel labeling cost.

The graph $G$ has nodes $s$, $t$ and a node for every image pixel, see Fig. 1.6(a). The unary potentials in (1.20) are encoded as edges between the image nodes, and the $s$ and $t$ nodes. If $p$ is an image node, there will be $(t,p)$ and $(p,s)$ edges with costs $D_{p}(s)$ and $D_{p}(t)$, respectively, see Fig. 1.6(b). The pairwise potentials in (1.20) are encoded as edges between the image nodes with cost $\lambda$. The example shown in Fig. 1.6(c) connects every image node, except the boundary nodes, to its 4 nearest neighbours. Figure 1.6(d) shows the complete constructed graph.

Figures 1.6(e) and (f) show a possible $s\backslash t$-cut and its corresponding pixel labeling, respectively. The cost of the $s\backslash t$-cut shown in Fig. 1.6(e) is the sum of the costs of the gray edges. A proof by contradiction could be derived to show that for an $s\backslash t$-cut, e.g. Fig. 1.6(e), there is a unique pixel labeling corresponding to it, e.g. Fig. 1.6(f), and that their costs are equal.

In terms of complexity, the max-flow of $G$ could be found in polynomial time [1] and hence its dual solution, i.e. the min-cut, as well. Also, the min-cut will naturally lead to a unique partitioning of the image pixels, i.e. its corresponding pixel labeling. We will use the term graph-cuts to refer to min-cut/max-flow based algorithms used for labeling problems.

In contrast to the two labels case, when there are three or more labels the labeling problem becomes $\text{NP}$-hard [4, 22]. The $\alpha$-expansion algorithm [4] is one of the graph-cut based algorithms that could be used to find an approximate solution for (1.20). Other than the graph-cut algorithms [4] there are other different optimization frameworks that could be used to find an approximate solution for energy (1.20), e.g. belief propagation [23] and dual decomposition [11].

\textbf{Case $|\mathcal{L}| \geq 3$:} The main idea of the $\alpha$-expansion algorithm [4] is to break the multi-labeling problem, with more than two labels, into a series of binary labeling problems each of which could be solved by a single graph-cut. These binary labeling problems are referred to as “\textit{expansion moves}” because label $\alpha$ is given a chance to grow arbitrarily.

The $\alpha$-expansion algorithm maintains a current labeling $f'$ and iteratively ‘moves’ to a better one until no improvements can be made. At each iteration, some label $\alpha$ is randomly chosen and variables $f_p$ are simultaneously given a binary choice either to stay as $f_p = f'_p$ or switch to $f_p = \alpha$. An expansion move is accepted and the current labeling is updated if the new labeling decreases the energy. The algorithm terminates when there
is no expansion move that would decrease the energy anymore. It should be noted that
an expansion move builds a graph similar to the binary s-t-cut case but with additional
auxiliary nodes to properly account for the smoothness costs, see [4] for details.

In contrast to standard moves as in Iterated Conditional Modes [2] and annealing
[9] where only one pixel is allowed to change its labeling, expansion moves allows “any”
subset of image pixels to change their labels to $\alpha$. In addition, a local minimum generated
by $\alpha$-expansion for energy (1.20) is within a factor of 2 of the global minimum [4].

1.3 Thesis Overview

1.3.1 Basic Problem Formulation

This section describes our formulation of matching and multi-model fitting problems and
some basic notations. For simplicity of presentation, in this section we consider only a
simple type of model, i.e. translation. In general, our formulation also applies to other
model types, e.g. planes, homographies etc., as explored in later chapters.

Let $l$ and $r$ be a pair of (left and right) images where different corresponding parts
undergo different translations. Since we are dealing with a discrete energy we can only
consider a finite set of translations (models) with a corresponding set of indices $L = \{1, \ldots, K\}$. The example in Fig. 1.7(a) shows images $l$ and $r$ where corresponding parts
have the same color. First we extract image features $F_l$ and $F_r$, e.g. SIFT [13], from $l$ and $r$, respectively. Extracted features are shown as black dots in Fig. 1.7(b).

We need to find a one-to-one matching between feature sets $F_l$ and $F_r$. Such a
matching can be described by a set of boolean variables $M = \{x_{pq} | \forall p \in F_l, q \in F_r\}$, e.g. see Fig. 1.7(b), and a set of linear constraints standard in assignment problems [5]

$$
\sum_{p \in F_l} x_{pq} = 1 \quad \forall q \in F_r
$$
$$
\sum_{q \in F_r} x_{pq} = 1 \quad \forall p \in F_l
$$
$$
x_{pq} \in \{0, 1\} \quad \forall p \in F_l, \forall q \in F_r.
$$

The multi-model fitting problem is to identify translations (model parameters) and
to assign them to matches. We introduce two sets of variables: models' parameters $\theta = \{\theta_h | h \in L\}$ and labeling $f = \{f_p \in L | \forall p \in F_l\}$. For the example in Fig. 1.7, an
ideal set of model parameters should include translations $\theta_i$ and $\theta_j$ that the green and
blue parts undergo, see Fig. 1.7(c). The ideal labeling $f$ assigns model indices $i$ and $j$ in $L$ to features in green and blue parts respectively, see Fig. 1.7(d). For convenience,
we assign model indices $f_p \in L$ to features in one of the images rather than to matches.
This is valid because each feature $p \in F_l$ corresponds to a unique match.

Now that our variables are introduced, we can discuss our joint fitting and matching
objective function

$$
E(M, \theta, f) = \text{Data Term} + \text{Label Cost}.
$$

Our data term consists of matching and fitting costs, i.e. an appearance penalty for
matching $p \in F_l$ to $q \in F_r$ and a fitting penalty for assigning the resulting match to
model $\theta_{f_p}$. The appearance penalty encourages matching visually similar features. The
fitting penalty prefers smaller geometric error when assigning models to matches. The label cost term in (1.21) is a sparsity regularizer that penalizes the number of distinct labels/models that are assigned to matches. Label cost is needed to avoid over-fitting, e.g. a trivial solution where every match is assigned a distinct model that perfectly fits it.

Minimizing energy (1.21) is an NP-Hard problem and we propose an approximation algorithm. Furthermore, when matching $\mathcal{M}$ is known energy (1.21) reduces to the multi-model fitting problem with objective function

$$E(\theta, f) = \text{Data Term} + \text{Label Cost} \quad (1.22)$$

which is also an NP-Hard problem. We will discuss uncapacitated facility location (UFL) based approximation algorithm for energy (1.22). Moreover, for multi-model fitting problem we propose and test a more general regularization framework

$$E(\theta, f) = \text{Data Term} + \text{Spatial Smoothness} + \text{Label Cost} \quad (1.23)$$

which incorporates spatial smoothness regularizer to encourage neighbouring features to be assigned to the same label/model.

1.3.2 Summary of Contributions

Feature matching and model fitting are fundamental problems in multi-view geometry. However, they are chicken-&-egg problems. On the one hand, if the matches are known a single model could be easily estimated using standard techniques, e.g. RANSAC [8], MLESAC [20] etc. On the other hand, knowing the model could significantly simplify the matching, e.g. if the fundamental matrix is known for a stereo pair we only need to match features along corresponding epipolar lines as in guided-matching [10].

Standard multi-view geometry techniques sequentially solve feature matching and model fitting as two independent problems. One major drawback of that approach is the that if the matching step results in a bad solution, the fitting step would fail. Standard matching methods rely on the discriminative power of feature descriptors when identifying the matches. In scenes with repetitive textures or wide baseline the discriminative power degrades and standard matching methods fail; due to matching ambiguity a large number of true positive matches are discarded in order to avoid high false positive rates.

Standard model fitting methods [19, 12, 8, 24, 20] assume fixed feature matches. These methods could be classified into single and multi model fitting methods. In case of single model fitting, typically a model is selected greedily by maximizing the number of inliers within some fixed threshold. This idea was popularized by RANSAC [8] and this approach works well provided that data actually supports a “single” model. However, when data supports multiple models the fitting problem becomes challenging.

The current state-of-the-art multi-model fitting methods commonly use heuristics. For example, sequential-RANSAC iteratively finds and removes the model with largest support and this could inhibit the detection of subsequent models with smaller support. Schindler et al. in [16] assume that each match is an inlier for no more than 2 proposed
Figure 1.7: (a) shows a pair of images with parts (green and blue) that undergo different translations. (b) shows feature sets $\mathcal{F}_i$ and $\mathcal{F}_r$, and a possible match. (c) shows translations $\theta_i$ and $\theta_j$ that the green and blue parts undergo. (d) shows an ideal labeling: green’s part features (inside green cloud) are labeled $i$ and blue’s part features (inside blue cloud) are labeled $j$. 

(a) Pair of images

(b) Features $\mathcal{F}_i$, $\mathcal{F}_r$ and a possible match

(c) Translations $\theta_i$ and $\theta_j$

(d) Ideal labeling
1.3. Thesis Overview

models. This strong assumption requires a pre-processing data analysis step that prunes
the set of initially sampled models leaving only a relatively small set of good candidates.

One of the main contributions of this thesis is a joint formulation of fitting and
matching problems. We are first to introduce an objective function combining both
matching and multi-model estimation. Our approach considers both features appearances and geometric fitting errors. We also propose an approximation algorithm for the
corresponding NP-hard optimization problem that uses block-coordinate descent with
respect to matching and model fitting variables. For fixed models, our method uses
an efficient min-cost-max-flow based algorithm, i.e. Local Search-GAP, to solve a novel
generalization of the linear assignment problem with label cost, i.e. Regularized-GAP. For
fixed matching, the problem reduces to multi-model fitting problem for which we propose
an approximation algorithm, PEARL.

In contrast to existing methods [6, 17], our joint fitting and matching approach is
not model specific as it could be used to fit homographies, rigid-motions, fundamental
matrices, etc. Moreover, it is specifically designed to simultaneously fit multiple-models.
Unlike [21], our method could efficiently match a large number of features. Experimental
results show that our approach significantly increases the number of correctly matched
features, improves the accuracy of estimated models, and is robust in cases with large
baselines or repetitive texture. In particular, our method could be used for 3D recon-
struction to eliminate the need for a large number of small baseline image pairs\(^5\) which
would significantly reduce the computational load.

We also introduce global objective functions for multi-model fitting with fixed matches
that uses various forms of regularization (spatial smoothness and sparsity) and propose
a graph-cut based optimization algorithm, PEARL. Our energy functional balances geo-
metric errors against spatial coherence of matched features while penalizing the number
of labels/models assigned to these matched features to avoid over-fitting. In contrast to
state-of-the-art, our energy formulation utilizes two regularizers, namely spatial smooth-
ness and label cost (sparsity). The spatial regularizer penalizes discontinuous, i.e. neigh-
bouiring features are encouraged to be assigned to the same label/model. The label cost
enables our energy to prefer solutions where fewer distinct models (labels) are assigned
to matches. We were able to achieve high quality results for multi-homography fitting,
multi-motion detection and various other applications.

List of Contributions:

- Joint multi-model fitting and matching objective function and an approximation
  algorithm to optimize it.

- Regularized-GAP, a novel generalization of the linear assignment problem incorpo-
  rating label cost, i.e. sparsity prior.

- An efficient min-cost-max-flow based heuristic algorithm, Local Search-GAP, to
  optimize Regularized-GAP.

\(^5\)Needed to propagate matches between image pairs with large baselines.
• Multiple regularization forms for the multi-model fitting problem with fixed matches and a graph-cut based optimization algorithm to optimize them, PEARL.

• A graph-cut algorithm extending $\alpha$-expansion [4] to multi-labeling problems with spatial smoothness and label cost, i.e. energy (1.23). PEARL either uses $\alpha$-expansion or its proposed extension as a sub-procedure depending on the choice of regularization terms in the objective function.

1.3.3 Thesis Structure

In Chapter 2 we introduce various forms of regularization for the multi-model fitting problem with fixed matches and a graph-cut based optimization algorithm to optimize them, PEARL. In Chapter 3 we propose an extension to $\alpha$-expansion [4] to optimize multi-labeling problems with spatial smoothness and label cost, i.e. energy (1.23). Finally, in Chapter 4 we propose the joint formulation of multi-model fitting and matching problems and an approximation algorithm for the corresponding $\text{NP}$-hard optimization problem. The multi-model fitting method covered in Chapter 2 is used as a sub-procedure in one of the steps of the proposed block coordinate descent approach for joint fitting and matching discussed in Chapter 4.

Bibliography


Chapter 2

Energy-based Geometric Multi-Model Fitting

Geometric multi-model fitting is a typical chicken-&-egg problem: data points should be clustered based on geometric proximity to models whose unknown parameters must be estimated at the same time. Most existing methods, including generalizations of RANSAC, greedily search for models with most inliers, within a threshold, ignoring overall classification of points. We formulate geometric multi-model fitting as a multi-labeling problem with a global energy function balancing geometric errors and regularity of inlier clusters. Regularization based on spatial coherence (on some near-neighbor graph) and/or label costs is NP-hard. The combinatorial algorithm introduced in Chapter 3 can minimize such regularized energies over a finite set of labels, but it is not directly applicable to a continuum of labels, e.g. \( \mathbb{R}^2 \) in line fitting. Our proposed approach PEARL combines model sampling from data points as in RANSAC with iterative re-estimation of inliers and models' parameters based on a global regularization functional. In practice, PEARL converges to an approximate but good quality solution, shown empirically, of our proposed energy while automatically selecting a small number of models that best explain the whole data set. Our tests demonstrate that our energy-based approach significantly improves the current state-of-the-art in geometric model fitting currently dominated by various greedy generalizations of RANSAC.
Chapter 2. Energy-based Geometric Multi-Model Fitting

Figure 2.1: Blue dots are data points supporting 8 lines. In multi-model cases, detecting models by maximizing the number of inliers may work for low levels of noise (a). Higher noise levels require larger thresholds to detect inliers (b), but then some random model (red) may have far more inliers than the true model (green). The integers show the number of model inliers for the selected thresholds. Simplistic greedy selection of models with the largest number of inliers would fail in (b). This example illustrates a general problem that affects many multi-model fitting approaches that greedily selects one model at a time while ignoring the overall solution. It also motivates our global energy approach that considers the quality of the whole solution.

2.1 Introduction

We study a general case of geometric multi-model fitting problem where data is a mixture of outliers and points supporting an unspecified number of models of some known type\(^1\). The majority of existing algorithms treat inlier classification and parameter estimation as two independent problems. Typically, each model is selected greedily by maximizing inliers within some fixed threshold. This approach works well when data supports a single model [14], but we argue that this approach is fundamentally flawed in multi-model cases, see Fig. 2.1.

\textsc{RANSAC} [14] is a well-known robust method for dealing with large number of outliers when the data supports only one model. The main idea is to generate a number of model proposals by randomly sampling data points to estimate these model proposals and then selecting the model with the largest set of inliers (a.k.a. consensus set) with respect to some fixed threshold. Many publications [28, 33, 40] proposed various generalizations of \textsc{RANSAC} for multi-model fitting. For example, [28, 33] run \textsc{RANSAC} sequentially. Each iteration of these methods selects one of the randomly sampled models that maximizes either the number of inliers or some similar threshold-based measure. Then the identified model’s inliers are removed from the set of data points before the next iteration looks for the next model. Other methods rely on different forms of greedy clustering, e.g. J-linkage.

\(^1\)For simplicity, we assume (parametric) models of the same type. This is not essential.
2.1. Introduction

(a) fitting homographies (stereo) (b) estimation accuracy vs. \( \lambda \)

Figure 2.2: Motivating spatial regularization in geometric model fitting. In many vision problems combining geometric errors and spatial coherence terms in energy (2.3) can be justified \textit{generatively} because clusters of inliers are generated by regular objects (a). Moreover, spatial regularization can also be justified \textit{discriminatively}. Plot (b) shows an average deviation from the ground truth for optimal models obtained in 100 randomly generated line-fitting tests as in Fig. 2.4(d). Each point in this plot corresponds to some fixed smoothness parameter \( \lambda \) in (2.3). Clearly, the spatial coherence term significantly reduces estimation errors for some \( \lambda > 0 \).

[26], implicitly maximizing the number of inliers within a given threshold. One can also apply Hough transform to formulate multi-model fitting as a clustering problem in the space of model parameters and use \textit{mean-shift} [9] to identify the modes in this Hough space. It is easy to see that this approach also greedily maximizes the number of inliers.

We argue that, in general, greedy selection of one model while ignoring the overall solution is a flawed approach to multi-model cases. Figure 2.1 shows a simple example illustrating the typical problem in the context of greedy inlier maximization: if the level of noise is increased, some random model can have a larger number of inliers than the true models. This also explains our results in Section 2.3 and Figs. 2.7-2.9 which show that many existing greedy methods work only on examples with low levels of noise and clutter.

2.1.1 Towards Energy Optimization

This thesis argues that geometric multi-model fitting is better formulated as an optimization problem with a global energy functional describing the quality of the overall solution. An energy function sets some specific “goodness” criterion for different solutions and the sought optima can be seen as “objectively” the best solution with respect to this criterion. On the one hand, there are many problems in computer vision (e.g. segmentation, optical flow, stereo) routinely solved as optimization problems with global energy functionals. On the other hand, there are a few examples in vision [30, 28, 2, 19] where some specific geometric multi-model fitting problems were approached using a global energy
functional. Perhaps, the limitations of these methods which we describe in Section 2.1.2 explain why many researchers in the community still prefer greedy heuristics. Our goal is a general energy-based framework for geometric multi-model fitting problems with efficient algorithms and wide applicability in computer vision.

There are several limitations for using standard energy-based methods for mixture models, such as EM or K-means, in geometric multi-model fitting problems in vision. In general, these methods may not be robust to outliers and noise. They are only guaranteed to find a local minimum and are known for sensitivity to initialization, e.g. see [28] and a detailed discussion in [13] Section 3.2. Models should be represented as probability distributions in EM, which is not always straightforward in geometric problems in vision. The standard versions of EM and K-means do not address spatial regularity explicitly. There are extensions of EM regularizing the number of models, e.g. using Dirichlet prior [3]. In the context of Gaussian mixture models (GMM), [13] combine Dirichlet sparsity prior with a large number of initial proposals, which is shown to better avoid local minima. In practice, the algorithm in [13] changes the energy functional when removing each redundant weak model. To achieve a sufficiently strong model pruning effect, one should also use improper negative values of Dirichlet distribution parameter as in [13] and [11] (Fig. 12). Both K-means and EM are more common in problems with a fixed number of models. For example, the EM framework in MLESAC [29] is fixed to 2 models (inliers/outliers), and the method in [15] estimates a known number of motions in cases with relatively low noise\(^2\). Soft assignment of inliers is an advantage of the EM algorithm in solving general mixture problems (e.g. GMM) where models can spatially overlap, but this may not be useful in geometric problems in vision where models typically have distinct spatial support, see Fig. 2.2(a). Standard K-means is also known to have a bias towards equally dividing the points among the models.

In order to motivate our general approach, we first demonstrate some energy-based interpretation of the basic RANSAC algorithm [14]. This interpretation is limited to a simple case when data supports only one model (e.g. one line). Let \( p \) be a data point in the data set \( \mathcal{P} \), and \( f_p \) be a boolean identifying whether \( p \) is an inlier or not. Also, let \( f = \{ f_p \mid p \in \mathcal{P} \} \). The main goal of RANSAC is to find parameters \( \theta \) of the model with the largest number of inliers within some threshold \( T \). And, also to identify whether each data point \( p \) is an inlier or not, i.e. find \( f \). This can be represented as minimization of energy

\[
E(\theta) = \sum_{p \in \mathcal{P}} ||p - \theta||,
\]

where

\[
||p - \theta|| = \begin{cases} 
0 & \text{if } \text{dist}(p, \theta) < T \\
1 & \text{otherwise}
\end{cases}
\]

and \( \text{dist}(p, \theta) \) is Euclidean distance between data point \( p \) and the nearest point on model \( \theta \). Note that RANSAC optimizes \( E(\theta) \) only over model parameters \( \theta \) and the inliers are identified implicitly from the threshold \( T \), i.e. \( f_p = 1 \) if \( \text{dist}(p, \theta) < T \) and 0 otherwise. RANSAC’s energy \( E(\theta) \) counts the outliers for \( \theta \) using the 0-1 measure of \( ||p - \theta|| \) above. Note that the standard RANSAC algorithm is a heuristic for maximizing the number

\(^2\)From personal communications with A. Gruber.
2.1. Introduction

Figure 2.3: Examples of inlier classification from thresholding. If data is known to support one model (a) then thresholding identifies inliers (blue) for any model without ambiguity. In case of 3 models (b), simple thresholding may not disambiguate inliers (red) between the models.

of inliers, but in some cases it is possible to find the global optimum [23, 39]. Standard RANSAC also utilizes a post processing step that refines the model’s parameters $\theta$ by minimizing the sum of squared errors of the identified inliers. Thus, a more principled optimization-based formulation of RANSAC leads to MLESAC energy [29] using truncated Euclidean errors

$$||p - \theta|| = \begin{cases} dist^2(p, \theta) & \text{if } dist^2(p, \theta) < T \\ T & \text{otherwise.} \end{cases}$$

Now assume that data supports multiple models, say $k$ models. Let $\mathcal{L}$ denote the set of model indices $\mathcal{L} = \{1, 2, \ldots, k\}$. It is possible to formulate geometric multi-model model-fitting as optimization of energy $E(\theta_1, \theta_2, \ldots, \theta_k)$ over $k$ models’ parameters. As in the earlier example with a single model, this approach needs some implicit assignment of inliers to models. In the multi-model case, however, this could be non-trivial. As shown in Fig. 2.3(b), simple thresholding may assign a point to several models. Interestingly, the EM framework for mixture models [3, 29, 15] corresponds to energy $E(\theta_1, \theta_2, \ldots, \theta_k)$. EM uses implicit “soft” classification of inliers computed in an intermediate optimization step. Even though the standard version of EM algorithm needs the number of models to be known, there are many generalizations of EM that could be worth studying in the context of geometric applications in vision. However, we prefer to focus on a fairly different energy formulation based on explicit “hard” classification of inliers. As shown in Fig. 2.2(a), in many problems in computer vision geometric models have non-overlapping spatial support, which better corresponds to hard assignment of inliers.

We formulate geometric multi-model fitting in the general case when the data supports an unknown number of models. In the multi-model case, each data point $p$ is assigned-(identified as an inlier) to one model, i.e. $f_p \in \mathcal{L}$. Also for simplicity let $\theta$ denote the set of models’ parameters, $\theta = \{\theta_h \mid h \in \mathcal{L}\}$. Model fitting could be formulated as minimization of energy $E(f, \theta)$ over labeling $f = \{f_p \mid p \in \mathcal{P}\}$ of data points and models’ parameters $\theta$. If the goal is only to assign data points to models given the model parameters, as in our one-model example, then $E(f, \theta)$ reduces to the following multi-labeling problem

$$E(f) = \sum_{p \in \mathcal{P}} ||p - \theta_{f_p}||$$  (2.1)
where $||p - \theta_{f_p}||$ is the fitting error measure. Obviously, this functional would not work well as the optimal solution will independently fit some model $f_p$ to each point $p$. This corresponds to over-fitting: every point is assigned some perfectly fit model and there are no outliers. It is clear that model fitting/labeling errors (2.1) must be combined with some energy term regularizing the labeling.

One form of regularization for (2.1) could be to fix the number of allowed distinct models/labels. Then, energy (2.1) corresponds to the standard K-means algorithm. This approach, however, does not work if the exact number of models is not known a priori. Recently, Li [19] proposed a soft form of regularization for the number of models by combining geometric errors with the label count penalty

$$E(f) = \sum_{p \in \mathcal{P}} ||p - \theta_{f_p}|| + \beta \sum_{h \in \mathcal{L}} \delta_h(f)$$

(2.2)

where $\delta_h(f) = [\exists p \in \mathcal{P}|f_p = h]$. Ten years earlier Torr [28] suggested even more general form of such regularization where each distinct model (label) gets a penalty defined by the model’s complexity instead of some fixed constant $\beta$. This approach allows to fit models of different types. In general, geometric model-fitting using energies like (2.2) is a very interesting idea, but specific algorithms for minimizing such energies proposed in [28] and [19] are fairly limited, see Section 2.1.2. We also argue that spatial regularity of inliers is required in many typical vision problems, see Fig. 2.2(a).

We propose two specific general forms of regularization in the context of inlier classification for geometric model fitting. In particular, we consider spatial regularization, i.e. energy (2.3)

$$E(f) = \sum_{p \in \mathcal{P}} ||p - \theta_{f_p}|| + \lambda \sum_{(p,q) \in \mathcal{N}} w_{pq} \cdot [f_p \neq f_q],$$

where $\mathcal{N}$ is some neighborhood (e.g. edges on some near-neighbor graph), and a more general combination of spatial smoothness with label counts, i.e. energy (2.7)

$$E(f) = \sum_{p \in \mathcal{P}} ||p - \theta_{f_p}|| + \lambda \sum_{(p,q) \in \mathcal{N}} w_{pq} \cdot [f_p \neq f_q] + \beta \sum_{h \in \mathcal{L}} \delta_h(f)$$

where [] are the Iverson brackets—if the condition inside the brackets is true the Inverson brackets return 1 and 0 otherwise.

While spatial regularization is very common in vision in general [2, 5, 34], it is not common in geometric model fitting. In part, this could be explained by the fact that spatial coherence is hard to justify generatively in applications where data points are i.i.d. samples. But in computer vision, see Figure 2.2(a), one can defend generative models of spatial regularity. Figure 2.2(b) also suggests that spatial regularization may work discriminatively even for i.i.d. data.

---

3One can not expect spatial regularization to work well for i.i.d. data, in general. Line fitting examples in Section 2.2 are a special case where it does work for i.i.d. points. We use these line fitting examples only to illustrate the basic operations of our algorithm. The primary target of our model fitting approach are applications in vision Section 2.3 where spatial coherence is well-founded.
This work demonstrates the significance of efficient combinatorial optimization methods for a wide range of geometric applications in computer vision. Surprisingly, such methods are overlooked in geometric model fitting, even though they are very common in segmentation, dense stereo, optical flows, and other problems. We contribute a new approach and significant algorithmic ideas specific to general geometric multi-model fitting. We see our two main contributions as follows:

- We propose a general practical energy-based framework for geometric model fitting particularly suitable for a wide range of applications in vision. To the best of our knowledge, energies (2.3) and (2.7) were not used for general geometric model fitting problems in the past. We demonstrate conceptual advantages and significant practical improvements over the state-of-the-art methods on a large number of generic applications in vision (line/plane fitting, homography estimation, rigid motion detection). In particular, we argue against typical greedy heuristics currently dominating in geometric model fitting and hope that our work will encourage a wider use of energy optimization methods well known in other areas of computer vision.

- Energies (2.3) and (2.7) can be addressed by existing powerful combinatorial optimization techniques with guaranteed optimality bounds, e.g. $\alpha$-expansion [5], only in cases of finite sets of labels. This limits their use for geometric model fitting where the space of model parameters is a continuum, e.g. $\mathbb{R}^2$ in line fitting. We propose a practical method PEARL for efficiently exploring the continuum of labels (model parameters) in the context of energy-based geometric model fitting. PEARL may also be juxtaposed with the contribution in [13] where a multitude of initial proposals combined with a sparsity prior are shown to reduce sensitivity of EM to local minima. Likewise, we show that combining a large number of initial random proposals with combinatorial algorithms produces robust solutions for our discrete model fitting functionals.

Our general approach alleviates dependence of many previous geometric model fitting methods on thresholding. The proposed methods for optimizing energies (2.3) and (2.7) work quite differently from greedy selection of models by the largest number of inliers. Our approach is robust to high levels of noise and clutter. It tries to find the optimal set of labels/models with a good fit to data points.

In order to apply standard discrete optimization algorithms to energies (2.3) and (2.7), we generate a large number of proposed labels (models) by sampling data points as in RANSAC. The goal of this step is to prune the search space (the continuum) of model parameters. As in RANSAC, the number of sampled models should be sufficiently large to guarantee with some level of confidence that at least one sample was generated from inliers for each true model. Such a finite pool of labels is likely to contain good model candidates. However, in contrast to RANSAC-style methods we rely on optimization of a global energy functional to select a small subset of models from this large (but finite) pool of proposals. Exploration of the continuum of labels (model parameters) is further significantly improved by iterating inlier segmentation for a finite set of labels and re-estimation of these labels (model parameters) from their inliers. These steps correspond
to a block coordinate descent of $E(f, \theta)$ over $f$ and $\theta$, respectively. As empirically shown in Fig. 2.6, the iterative refinement of models’ parameters and inlier classification of our approach allows it to find better solutions even when using a small set of initially proposed models.

### 2.1.2 Related Work

Our work proposes, justifies, and validates a wide class of regularization energies and a powerful iterative optimization technique as a general framework for geometric multi-model fitting particularly suitable in vision. Other geometric model fitting works have used separate elements of our approach such as RANSAC-style random sampling [28, 19] or EM-style iterations [2], but none have combined them in a single optimization framework. We also use a more general form of regularization (2.7) than any earlier geometric fitting methods. Our experiments show that our general energy-based approach works better than many state-of-the-art algorithms in this area. In other settings (segmentation, stereo) some elements of our framework have been used in various application-specific ways [38, 2, 24, 37].

Probably the earliest efforts to formulate an energy-based framework for geometric model fitting in vision is due to [30]. They optimize a likelihood function over binary indicator variables associated with a multitude of proposed models under the uniqueness constraint: each data point could be an inlier for at most one model. The corresponding integer programming problem is solved with a generic branch-and-bound solver. Another early paper by Torr [28] carefully justifies a version of model-fitting energy like (2.2) from an information criterion. The specific optimization technique used in [28] was EM initialized by a few models selected via sequential RANSAC\(^4\). As pointed out in [28], the solution generated by EM strongly depends on the quality of the initial models. Figures 2.1 and 2.9 show that greedy procedures like sequential RANSAC may be non-robust.

Variants of sequential RANSAC are also commonly used as a preprocessing step for dense MRF-based segmentation of image pixels with geometric labels. For example, Wills et al. [35] address the problem of finding large constant motions in optical flow imagery as follows. First, they use an extension of sequential RANSAC to detect a few rigid motions from pairs of matched discrete features. Then, they assign image pixels to these motions based on color consistency and Potts regularization using $\alpha$-expansion algorithm [5].

Some related MRF-based formulations used convergent iterative re-estimation of geometric models. Birchfield & Tomasi [2] estimate affine geometric models in a way specific to dense narrow base-line stereo. They combine photo-consistency of pixels with spatial regularization on a grid. Unlike [35], their initial geometric models are estimated from a disparity map generated by another stereo algorithm. The most noticeable overlap of our approach and [2] is the iterative use of $\alpha$-expansion and model re-estimation steps. After [2] and [24] such EM-style optimization became fairly common for different problems in vision. In contrast to [2], however, our framework is suitable for a significantly more general set of geometric problems. For example, instead of photo-consistency we

\(^4\)The actual term was introduced in [33] a few years later.
optimize geometric errors and combine them with more general forms of regularization, e.g. (2.7). Our method is more concerned with fitting to sparse data. Finally, we do not need to run other algorithms to initialize. Our experiments in Section 2.3.1 show noticeable improvement of accuracy on examples from [2].

Zabih & Kolmogorov [37] also use iterative optimization as in [2, 24] specifically in the context of image segmentation. They use standard spatial regularization like in (2.3) to cluster image pixels into spatially coherent segments with automatically estimated color models. The color models are initialized in an application specific way. In contrast, we work with a very different problem of geometric model fitting studying the more general label cost functional (2.7). In fact, our recent work [11] with additional coauthors shows that energy (2.7) may significantly improve image segmentation results. Minimum description length (MDL) interpretation of (2.7) has been well known in segmentation literature for some time [18, 38].

Schindler and Suter [25] proposed a related optimization method for geometric model fitting based on an approximation of label cost functional (2.2) without the spatial regularity term. Similar to our work, the goal in [25] is to detect geometric models using some global energy optimization instead of greedy heuristics like sequential RANSAC. They formulate a quadratic pseudo-boolean optimization (QPBO) problem over indicator variables defined for models proposed from data. These binary variables are similar to those in [30]. However, instead of enforcing the exact uniqueness constraint [30] (see above), the energy formulation in [25] makes an assumption that each data point is an inlier for no more than two proposed models. This strong assumption requires a pre-processing data analysis step that prunes the set of initially sampled models leaving only a relatively small set of good candidates. The actual optimization over binary indicator variables for such candidate models is performed using a standard Taboo-search algorithm.

The paper by Li [19] is probably the most closely related prior work. Similar to [28, 25], it formulates general geometric model fitting functional (2.2) and studies it in the context of rigid motion estimation, which we also consider as one of the applications in Section 2.3.3. Instead of the greedy approach of [28], [19] uses LP relaxation of (2.2) which could be slow. To speed up the method, [19] uses several application specific heuristics to significantly prune the set of proposed models. More importantly, [19] does not guarantee any optimality of the discrete solution obtained after rounding and the quality of such optimization could be an issue. These problems do not allow [19] to use EM-style iterative optimization that, in our experience, can significantly improve model fitting results. A better optimization of energy (2.2) with some optimality guarantees is discussed in Chapter 3.

In this paper, we argue that (2.7) is generally a better energy for geometric model fitting problems in vision. We found that per-label regularization term proposed in [28, 19] is a practically useful addition to standard spatial regularization (2.3). Fig. 2.10 shows one illustrative example where penalty for using each distinct label encourages the merging of isolated clusters supporting nearly identical models. Similarly, the results on real motion detection sequences in Figure 2.21 fail to merge spatially isolated background clusters into one motion if label counts are not a part of the energy. To optimize the third term in energy (2.7) one can use a simple and fast merging step in combination with standard $\alpha$-expansion optimizing the first two terms in the energy. This merging
heuristic is discussed at the end of Section 2.2. Alternatively, in Chapter 3 we provide
an extension of α-expansion algorithm that automatically accounts for the third term in
(2.7)—by incorporating it into each expansion step as a high-order clique. This Chapter
demonstrates that our optimization framework for multi-model fitting using either (2.3)
or (2.7) is a significantly better alternative to greedy generalizations of RANSAC cur-
rently dominant in geometric model fitting problems in computer vision.

2.2 Our Approach (PEaRL)

In this section we describe our algorithm for geometric model fitting in details. For
simplicity, the main ideas are illustrated in the context of synthetic multi-line fitting
examples. Section 2.3 validates our approach for estimating affine transformations, ho-
mographies, and rigid motion models in the context of computer vision.

Our proposed regularized energy combines geometric fitting errors with a spatial
smoothness term (2.3) and a label count penalty (2.7). However, the graph-based opti-
mization methods that we used to minimize that energy can only handle a finite number
of models/labels, e.g. ≤ 10000 or so. In our case, the labels are models described by
n real-valued parameters. To restrict the huge search space of labels \( \mathbb{R}^n \), in the first
step we propose a finite set of plausible models (labels). In the next step each label is
expanded to find its spatial support, i.e. assign data points to models/labels. Once the
inliers are fixed, the models’ parameters can be re-estimated by minimizing the geometric
errors. Note that the energy is bounded from below, since the geometric errors, and the
normalization constants \( \lambda \) and \( \beta \) are all non-negative. Furthermore, both the expand
and the re-estimate steps can not increase the energy thus convergence is guaranteed.
One can also iterate over the propose step either by further sampling the data points, or
by generating some new proposed labels (models) from the currently supported models,
e.g. by merging them. Below we provide more details about our model-fitting algorithm
in the context of energy (2.3), most of these details also apply to energy (2.7).

2.2.1 Proposing Initial Labels

First, our method uses random sampling of data points to propose an initial finite set of
models of size \( N \) where each model is in \( \mathbb{R}^n \) and \( n \) is the number of parameters describing
each model, e.g. \( n = 2 \) for lines and \( n = 6 \) for affine models. The idea of generating models
by sampling the data points is borrowed from RANSAC [14]. The required number of
initial models \( N \) is one of the parameters of RANSAC-based methods. It depends on the
number of data points, the number of outliers, the number of models to be estimated, the
minimum number of points required to estimate each model, and on the desired level of
confidence in the final solution, i.e. more sampled models means that it is more likely a
good estimate was sampled. The exact analysis for the case of estimating a single model
is given in [14] while its adaptation to multi-model case is discussed in [40, 33]. Although,
the theoretical \( N \) as in [40] is usually very large, our experiments suggest that in practice
PEaRL often needs far fewer samples than the theoretical estimate due to its iterative
nature that significantly improve probability of an accurate model reconstruction from
rough initial guesses.
2.2. Our Approach (PEaRL)

- initial 25 model proposals
- models & inliers (iteration 1)
- models & inliers (iteration 2)
- at convergence (iteration 5)

Figure 2.4: Illustration of PEARL’s iterations. (a) proposals generated by random sampling, (b-d) re-estimation of models and their inliers after several iterations of expand and re-estimate steps for energy (2.3) or (2.7). Note that the algorithm may converge to good models even from a small set of rough guesses. In this example we did not use an outlier label, see Section 2.2.3, and an optimal set of lines in (d) “explains” all data points.

2.2.2 Energy Formulation

We will use the multi-line fitting setting to motivate and introduce our energy. In multi-line fitting one is presented with a set $P$ of data points, and it is required to estimate the least number of models/lines that describe that data set and to identify the inliers of each model, i.e. assign the data points to the estimated models. As described earlier we restrict the search over the models’ parameter space via random sampling $N \leq 10000$. Let $\theta = \{\theta_h \mid \theta_h \in \mathbb{R}^n, \forall h \in \mathcal{L}\}$ denote the set of models’ parameters, and $\mathcal{L} = \{1, 2, \ldots, N\}$ be the set of indices to these models ($n = 2$ for multi-line fitting). In order to classify data points we will introduce a labeling variable say $f_p$ for each data point $p \in P$. We also define a labeling $f$ of the whole data set as $f = \{ f_p \mid \forall p \in P \}$. 
We propose the following regularized energy for multi-model fitting,

\[
E(\theta, f) = \sum_{p \in P} ||p - \theta f_p|| + \lambda \cdot \sum_{(p,q) \in \mathcal{N}} w_{pq} \cdot [f_p \neq f_q] \tag{2.3}
\]

where [ ] are the Iverson brackets\(^5\).

The first term in (2.3) is the sum of geometric errors between each data point \(p\) and its assigned model \(\theta f_p\). For example, the line fitting examples in this section use “perpendicular distance” between a 2D point \(p = (x, y)\) and a line \(\theta h = (a, b)\)

\[
||p - \theta h|| = \left(\frac{|y - ax - b|}{\sqrt{a^2 + 1}}\right)^2
\]

which is the distance from \(p\) to the nearest point on line \(\theta h\). Robust (truncated) measures are also possible. For fixed models’ parameters, the first term \(||p - \theta f_p||\) in (2.3) corresponds to the negative log-likelihood \(-\ln \Pr(p | \theta f_p)\) when the energy is interpreted as an MRF-based posterior energy. Thus, the use of quadratic distance for \(||p - \theta f_p||\) is equivalent to assuming Gaussian distribution for errors. Clearly, the optimal labeling \(f\) for (2.3) depends on specific choice of geometric measure \(||p - \theta f_p||\).

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 could be based on a triangulation of points, see Fig. 2.5. Weights \(w_{pq}\) set discontinuity penalties for each pair of “neighboring” data points if they were assigned to different labels. For example, the synthetic line fitting examples in this section use weights \(w_{pq}\) inversely proportional to the distance between points \(p\) and \(q\) because closer points are a priori more likely to fit the same model

\[
w_{pq} = \exp\left(-\frac{||p - q||^2}{\varsigma^2}\right).
\]

In all of our synthetic line experiments \(\varsigma\) was constant and it was chosen heuristically\(^6\) to be 5. Also, for the experiments in Section 2.3 we used constant weights \(w_{pq} = 1\).

### 2.2.3 Outliers

In the context of multi-model fitting the term outlier may become somewhat philosophical. For example, Fig. 2.4(d) shows PEARL’s final solution with respect to energy (2.3) where a small set of lines explains all data points. In this case the word “outlier” is subject to an interpretation. In particular, one could use any specific “outlier criterion” to classify weak models, e.g. those with sufficiently small number of inliers. We used a different approach common in MRF-based literature. We introduce a special “outlier model” and the corresponding label \(\phi\) which is always present in the pool of labels when minimizing energy (2.3) or (2.7). Any point \(p\) assigned this label is considered an outlier. In contrast to real geometric models, label \(\phi\) has constant fidelity measure \(||p - \phi|| = \gamma\) for

\(^5\)If the condition inside the brackets is true the Iverson brackets return 1 and 0 otherwise.

\(^6\)There parameters could be learnt via cross validation etc.
2.2. Our Approach (PEaRL)

Figure 2.5: We use Delaunay triangulations of data points. K-nearest points or other techniques can be used as well, particularly for higher dimensional data. We did not observe much difference in practical performance.

all points \( p \in \mathcal{P} \). Intuitively, the outlier label corresponds to a “uniform” model. Typical weak models incur regularization penalties (smoothness and label costs) while explaining only a few points. Thus, the outlier label becomes a better alternative label for these points which otherwise would be assigned to some weak models. For example, compare the optimization results in Fig. 2.4(d) obtained without the explicit outlier label and the results in Figs. 2.7-2.9(f) where \( \phi \) was added into the pool of labels when minimizing energy (2.3).

2.2.4 Expand and Re-estimate Labels

In a nutshell, the expand and re-estimate steps minimize (2.3) in a block coordinate descent fashion. In the expand step, the models’ parameters \( \theta \) are fixed and this reduces energy (2.3) to

\[
\arg\min_f \sum_{p \in \mathcal{P}} ||p - \theta_{f_p}|| + \lambda \cdot \sum_{(p,q) \in \mathcal{N}} w_{pq} \cdot [f_p \neq f_q].
\]  

(2.4)

Energy (2.4) uses an MRF-based regularization scheme that is common in computer vision. There are various existing algorithms that could be used to find an approximate solution to (2.4). In our work we used the graph-cut based combinatorial algorithm \( \alpha \)-expansion [5] to find an approximate labeling \( f \). We can expand the models to find their spatial support, i.e. labeling \( f \). In this case, it is possible to interpret \( \alpha \)-expansions as a competition among model/labels for spatial support; models with the best-fit to data points are assigned to a large number of spatially coherent “inliers”, while most of the “erroneous” models get little to no inliers.
In the re-estimate step, the labeling $f$ is fixed and this reduces energy (2.3) to

$$\arg\min_{\theta} \sum_{p \in P} ||p - \theta_{f_p}||. \quad (2.5)$$

Note that the second term in (2.3) becomes constant when the labeling is fixed. Energy (2.5) could be rewritten as

$$\min \sum_{p \in P} ||p - \theta_{f_p}|| = \min \sum_{h \in \mathcal{L}} \sum_{p \in \mathcal{P}_h} ||p - \theta_h||$$

where $\mathcal{P}_h = \{p \mid f_p = h\}$. Clearly, we can minimize this expression by re-estimating parameters $\theta_h$ of each model $h \in \mathcal{L}$

$$\theta_h \leftarrow \arg\min_{\theta_h} \sum_{p \in \mathcal{P}_h} ||p - \theta_h||. \quad (2.6)$$

Note that the re-estimate step does not affect the second (smoothness) term in (2.3) unless two models became equal (parameter wise) after the re-estimation step. In that case, the smoothness energy will also decrease.

There are many known methods for optimizing the sum of geometric errors (2.6). The used optimization method depends on specific choice of the geometric measure. For example, the minimum sum of squares of orthogonal errors in our lines-fitting examples could be obtained using a standard closed-form solution. A large number of other examples of geometric or algebraic error measures and different numerical methods for optimizing them are widely discussed in the computer vision literature, e.g. see [16].

Figure 2.4(b) visualizes clusters of inliers and re-estimated models obtained via the expand (inlier classification) and reestimate (model parameters) steps. In some cases it could be useful to iterate the propose step as well. For example, new labels can be generated by merging or splitting clusters of inliers. One interesting example of “merging” is described in the context of the example shown in Fig. 2.10.

### 2.2.5 Algorithm and Its Properties

Both expansion (inlier classification) and re-estimation steps decrease energy (2.3). Thus, we can iterate over these steps until convergence, see Fig. 2.4 (b-d). We can stop the iterations when a new round of $\alpha$-expansion does not change the labeling $f$. As soon as the spatial support of the current models (labels) stops changing, re-estimation of the models (2.6) cannot improve the geometric error term. Furthermore, since (2.3) is bounded from below PEaRL is guaranteed to converge. PEaRL algorithm (Propose Expand and Re-estimate Labels) is summarized below:
2.2. Our Approach (PEaRL)

**PEaRL Algorithm**

**Propose:**
Randomly sample data to generate model proposals \( \theta \)

**Repeat** till convergence

**Expand:**
Given \( \theta \), run \( \alpha \)-expansion [5] for energy (2.4) to find labeling \( f \)

**Re-estimate Labels:**
Given \( f \), solve (2.6) to reestimate models’ parameters and update \( \theta \)
(optional) sample more or merge/split current models in \( \theta \)

Figure 2.6 gives an idea of how the accuracy of estimated models depends on the number of initial randomly sampled proposals \( |L| \). For simplicity, we generated synthetic data supporting only one line. This also allows juxtaposing PEARL with standard RANSAC. Both methods used the same initial set of randomly sampled model proposals. RANSAC basically selected the model with the largest number of inliers w.r.t. some fixed threshold. It was easy to tune energy (2.3) so that an optimal data labeling is binary with the following two labels: one line label and outlier label \( \varphi \). Depending on initialization, PEARL’s iterative *expand* and *re-estimate* steps (illustrated in Fig. 2.4) would converge to different solutions. Normally, larger sets of initial random proposals leads to better solutions. Also, PEARL may converge to more than one line/model, in which case we reported the error for the model with the largest support. In general, Fig. 2.6 shows that by minimizing energy (2.3) PEARL can converge to a line significantly better than the best line from the randomly sampled set. In contrast, RANSAC strongly relies on a larger number of initial samples to find accurate estimates. Some generalizations of RANSAC explicitly improve the initial proposals, e.g. by re-sampling the inliers of the strongest initial models [8]. Similar to RANSAC, their generalization to multi-model problems could be problematic.

Figures 2.7-2.9 compare PEARL to existing geometric multi-model fitting methods [40, 26] and mean-shift [9] which were discussed in the introduction. The synthetic multi-line examples were generated using different levels of Gaussian noise and different number of uniformly distributed outliers. The previous methods were tuned to get the best results for each specific level of noise and clutter. To demonstrate robustness of PEARL, in each test we tuned only one parameter \( \sigma \) in the geometric error measure \( ||p - \theta_h|| = -\ln G_\sigma(p - \theta_h) \) where \( G_\sigma(\cdot) \) is a Gaussian distribution function and \( p - \theta_h \) is the distance from point \( p \) to line \( \theta_h \). Note that PEARL obtains very similar results when parameter \( \sigma \) is automatically estimated as described in the context of the example shown in Fig. 2.11.

For PEARL and multi-RANSAC the data points were uniformly sampled while for J-linkage and mean-shift we used distance-based sampling which helps J-linkage and mean-shift algorithms. Sampling closer points increases the probability that the sampled model is closer to one of the peaks in the Hough transform.

Mean-shift and J-linkage have no constraints on the number of models they generate. Figures 2.7-2.9 show their strongest 7 models. Multi-RANSAC had to be given the exact number of models. Compared to mean-shift and J-linkage, PEARL finds a very small
Figure 2.6: PEARL vs RANSAC for synthetic data sets with 80% uniformly distributed outliers and 20% noisy inliers supporting one line model. In each test both methods used the same initial set of randomly sampled model proposals. X-axis is the number of initially proposed models. Y-axis shows estimation errors w.r.t. the parameters of the ground truth line model. The errors are averaged over 400 randomly generated tests. RANSAC basically selects the best model from the initial set of proposals. By iteratively minimizing global energy (2.3) PEARL can converge to a much better model than those in the initial set. In contrast, standard RANSAC strongly relies on a larger number of initial samples to find accurate estimates.

number of models giving with the optimal fit to the data. But, in addition to correctly identified true models, it can “hallucinate” a few models among outliers as in Fig. 2.4(d). Weak models can be automatically filtered out by setting a very conservative limit on the minimum number of inliers.

The results for standard methods in Figures 2.7-2.9 are consistent with our earlier observations in Figure 2.1, i.e. common greedy heuristics selecting one model with a large score (e.g. number of inliers) independently from the overall solution could be problematic in multi-model problems. As illustrated in Fig. 2.1, random models may have higher scores (more inliers) than the true ones. This explains why many standard methods work relatively well only for the low noise example as in Fig. 2.7.
2.2. Our Approach (PEaRL)

Figure 2.7: Comparing the results for fitting lines to noisy data points. The data points were perturbed with a low level of Gaussian noise ($\sigma = 0.01$) and 200 random outliers were added. Outliers represent 25% of the data.
Figure 2.8: Comparing the results for fitting lines to noisy data points. The data points were perturbed with medium Gaussian noise ($\sigma = 0.02$) and included 300 random outliers. Outliers represent 35% of the data.
2.2. Our Approach (PEaRL)

(a) The data points (500 outliers)
(b) The Hough transform of the data
(c) Multi-RANSAC result
(d) mean-shift result
(e) J-Linkage result
(f) PEARL result

Figure 2.9: Comparing the results for fitting lines to noisy data points. The data points were perturbed with high Gaussian noise ($\sigma = 0.025$) and 500 random outliers were added. Outliers represent 45% of the data.
Chapter 2. Energy-based Geometric Multi-Model Fitting

Figure 2.10: Intersecting lines example. Optimization of energy (2.3) may leave spatially isolated groups of inliers assigned to 2 models even if their parameters are infinitesimally close (b). Per-label costs in energy (2.7) solves this problem (c) but requires either an additional merging operation or an extension of \( \alpha \)-expansion—introduced in Chapter 3. This example motivates the use of the EM-style soft assignments of labels to points. However, in computer vision as will be shown in Sec. 2.3 we get occlusions (not intersections), which better motivates our “hard” assignments of labels.

As we discussed in the introduction, coherence between inliers is often a good assumption particularly for problems in vision. Figure 2.10 shows one typical example where this assumption could be violated. Clearly, one of the intersecting lines cannot be assigned a spatially connected group of inliers. Moreover, an optimal solution for energy (2.3) cannot merge two models with very similar parameters if their inliers are spatially separated as in Fig. 2.10(b). However, a simple post-processing step after each expansion step can merge separated groups of inliers with similar models, as in Fig. 2.10(c), if the “average optimal model”\(^7\) increases the sum of geometric errors by no more than some predefined threshold \( \beta \). In fact, this merging operation can be seen as an optimization step if the energy function gets an additional term penalizing the number of models/labels with non empty support

\[
E(\theta, f) = \sum_{p \in P} ||p - \theta_{f_p}|| + \lambda \cdot \sum_{(p,q) \in N} w_{pq} \cdot [f_p \neq f_q] + \beta \cdot \sum_{h \in L} \delta_h(f) \tag{2.7}
\]

where \( \delta_h(f) = [\exists p \in P | f_p = h] \). Similar merging operations were also used in [38] for a continuous version of this label cost energy. Instead of the merging heuristic, energy (2.7) can be also minimized using an extension of \( \alpha \)-expansion algorithm proposed in Chapter 3.

Figure 2.11 demonstrates another interesting feature of our optimization approach. Unlike many previous multi-model fitting methods [33, 40, 33] that use fixed thresholds, PEaRL can identify multiple models with different levels of noise. For example, this can be achieved as follows. Assuming that geometric errors for inliers correspond to Gaussian noise, one can set geometric error penalty \( ||p - \theta_h|| \) according to the negative logarithm

\(^7\)The model estimated from the merged group of inliers.


2.3. Experimental Results

This section validates our model fitting technique PEARL on multi-view reconstruction data sets supporting multiple models. Our experiments in Sections 2.3.1 and 2.3.2 used affine and homography models, respectively. Data points were obtained by matching SIFT [20] features on rectified image pairs in narrow-based stereo, and on uncalibrated wide-base pairs.

Figure 2.11: Fitting lines with different noise levels. The inliers in (a) were generated with different levels of Gaussian noise. 40% of the data are outliers. In (b) PEARL estimated labels combining geometric model parameters with unknown noise variances using error measure (2.8). Previous multi-model fitting methods use fixed thresholds to identify inliers, which would not work in this case.

of the normal distribution function

\[ ||p - \theta_h|| = -\ln G_\sigma(p - \theta_h) \]

where \( p - \theta_h \) is the distance from \( p \) to the assigned model \( \theta_h \). Here one assumes some known \( \sigma \) parameter corresponding to the distribution’s variance. If models come with unknown different levels of noise, one can estimate more complex models \( \hat{\theta}_h = \{\theta, \sigma_h\} \) combining geometric model parameters \( \theta_h \) with its corresponding unknown noise-level \( \sigma_h \). In this case one can use the following error measure

\[ ||p - \hat{\theta}_h|| = -\ln G_{\sigma_h}(p - \theta_h). \]  \hspace{1cm} (2.8)

Figure 2.11 shows that this approach can correctly estimate both geometric and statistical noise-level parameters for each model.
2.3.1 Estimating Multiple Affine Models

In this section we apply PEaRL to estimate affine transformations in the context of rectified narrow baseline stereo. We use SIFT [20] features as points of interest, since they are scale and rotation invariant. They are also partially invariant to illumination and 3D camera viewpoint changes. Matches between pairs of points in two images are found using exhaustive search along the corresponding scan line. In principle, it is possible to replace exhaustive search with "smarter" methods as in [1, 22].

We will use the notation $(x_l, y_l)$, $(x_r, y_r)$ to describe the coordinates of the image feature on the left image $p_l$ and right image $p_r$. The symbol $p$ denotes a pair of matching points $(x_l, y_l, x_r, y_r)$ and $\mathcal{P}$ is the set of matched pairs. Restricting the search for matching pairs to the corresponding epipolar lines corresponds to imposing an additional constraint $|y_l - y_r| < \epsilon$ for some small threshold $\epsilon$.

A planar homography has only three degrees of freedom for rectified stereo images. In this case the epipole $e = [1 \ 0 \ 0]^T$ is at infinity and the fundamental matrix can be written as

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

where $[e]_x$ describes the skew-symmetric matrix of the vector $e$. Following [12], a planar homography $H$ satisfies the following constraint $H^T F + F^T H = 0$. Then, it can be shown that any planar homography for a rectified stereo pair is an affine transformation

$$
A = \begin{pmatrix}
a & b & c \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix},
$$

with 3 degrees of freedom corresponding to parameters $a$, $b$, and $c$. This transformation can be uniquely identified from three matching pairs. We first generate a finite set of initial model proposals by randomly sampling three pairs of matching points and by computing parameters $a$, $b$, $c$ for the corresponding models.

One simplistic way to measure geometric error between matching pair $p = (x_l, y_l, x_r, y_r)$ and model $A$ is a non-symmetric (left) transfer error from $p_l = (x_l, y_l)$ to $p_r = (x_r, y_r)$

$$
||p - A|| = |A p_l - p_r|^2 = \Delta_x^2 \tag{2.9}
$$

where

$$
\Delta_x = (ax_l + by_l + c - x_r)
$$

is a horizontal shift between $A p_l$ and $p_r$ along the epipolar line, see Fig. 2.12(a). This basic approach assumes that the vertical shift between $A p_l$ and $p_r$

$$
\Delta_y = (y_l - y_r)
$$

is zero. This could be justified because our matched pairs $p = \{p_l, p_r\}$ are points on the same scan lines ($|y_l - y_r| < \epsilon$) and affine transformation $A$ respects such (epipolar) lines.

---

8These are scan lines for rectified stereo images.
Figure 2.12: Geometric fitting errors $||p - A||$ for rectified stereo. Assuming matched points $p = \{p_l, p_r\}$ are on the corresponding epipolar (scan) lines $\Delta_y = y_l - y_r = 0$, the left transfer error is a horizontal shift $\Delta_x$ between $p_r$ and $Ap_l$ as shown in (a). The standard re-projection error (b) is obtained by treating points $p_l$ and $p_r$ as noisy observations of some hidden “true” pair of matched points $\bar{p} = \{\bar{p}_l, \bar{p}_r\}$ where $\bar{p}_r = A\bar{p}_l$ and $\bar{p}$ minimizes the observation noise $\min |\bar{p}_l - p_l|^2 + |\bar{p}_r - p_r|^2$. 

(a) non-symmetric (left) transfer error

(b) reprojection error
Alternatively, one can use the *re-projection error* \([16]\) illustrated in Fig.2.12(b). This approach treats \(p_l\) and \(p_r\) as noisy observations of some unknown “true” points \(\bar{p} = \{\bar{p}_l, \bar{p}_r\}\) estimated by minimizing the observation noise, as follows

\[
\|p - A\| = \min_{\bar{p}} |\bar{p}_l - p_l|^2 + |\bar{p}_r - p_r|^2 \quad \text{s.t.} \quad \bar{p}_r = A\bar{p}_l.
\]

For \(\bar{p} = \{p_l, Ap_l\}\) the objective function above equals \(|Ap_l - p_r|^2\) and, therefore, optimization over all \(\bar{p}\) should give an error smaller than (2.9). That is, our transfer error (2.9) can over-estimate the observation noise, as defined by the constrained optimization problem above. The difference could be particularly significant if plane \(A\) is near-horizontal.

Note that the optimal “true” pair \(\bar{p}\) corresponding to the minimum observation error is typically located on a scan line different from those containing data points \(p_l\) and \(p_r\).

If \(p_l\) and \(p_r\) are treated as noisy observations, these points do not need to respect the epipolar geometry. Thus, we can drop the constraint \(\Delta_y = 0\) when using the re-projection error. Following the definition in the previous paragraph, one can derive the following closed formula for the re-projection error specific to our affine transformations

\[
\|p - A\| = \frac{2\Delta_x^2 + (1 + a^2 + b^2)\Delta_y^2 - 2b\Delta_x\Delta_y}{2a^2 + b^2 + 2}.
\]

Figures 2.13(a-c) compare affine model fitting results regenerated by BT [2] and results generated by PEARL for two different geometric error measures \(\|p - A\|\) in (2.9) and (2.10). We applied PEARL to energy

\[
E(A, f) = \sum_{p \in P} \|p - Af_p\| + \lambda \sum_{(p,q) \in \mathcal{N}} [f_p \neq f_q] + \beta \sum_{h \in \mathcal{L}} \delta_h(f)
\]

where \(A = \{A_h \mid h \in \mathcal{L}\}\) is set of affine models. Our neighborhood graph \(\mathcal{N}\) is a triangulation of points \(p_r\) in the right image. BT [2] uses dense segmentation of pixels based on photo-consistency. This measure does not work well in texture-less regions and they have to rely on intensity edges (static cues) to detect the boundaries between regions supporting different models. In contrast, PEARL labels a sparse set of distinct features based on geometric errors and spatial regularization. Also, the label cost allows our method to connect spatially disconnected parts of the same model as in Fig. 2.10.

Our results in Figures 2.13(b-c) demonstrate that specific choice of geometric measure \(\|p - A\|\) can significantly change the results. For example, minimizing the horizontal transfer error in equation (2.9) worked well for all vertical planes in the scene, but it split the ground plane into two, see Fig. 2.13(b). The re-projection errors (2.10) worked significantly better than the transfer errors. This is particularly obvious for the ground plane. As mentioned earlier, the transfer error (2.9) significantly over-estimates the observation noise for near-horizontal planes. This is fairly analogous to the consequences of using “vertical” point-to-line distance \(\|p - L\| = |y - ax - b|^2\) for fitting near-vertical lines \(L = (a, b)\) to 2D points \(p = (x, y)\).

In order to provide some quantitative comparison between the affine models generated by BT [2] and PEARL, we used a “ground truth” image, shown in Fig. 2.14(a), where

\(^9\)Except when \(A\) is an exactly vertical plane.
2.3. Experimental Results

Figure 2.13: Results for "Clorox" stereo pair [2]. (a) Dense pixel segmentation by BT [2] uses photo-consistency. (b-c) Sparse inlier classification by PEARL using geometric fit measures (2.9) and (2.10), respectively.

<table>
<thead>
<tr>
<th>Line</th>
<th>BT [2]</th>
<th>PEARL (2.9)</th>
<th>PEARL (2.10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>34.55</td>
<td>8.80</td>
<td>4.37</td>
</tr>
<tr>
<td>$L_2$</td>
<td>16.83</td>
<td>4.82</td>
<td>7.5</td>
</tr>
<tr>
<td>$L_3$</td>
<td>5.56</td>
<td>13.27</td>
<td>9.53</td>
</tr>
<tr>
<td>$L_4$</td>
<td>5.99</td>
<td>4.46</td>
<td>8.47</td>
</tr>
<tr>
<td>Total</td>
<td>62.93</td>
<td>31.35</td>
<td>29.87</td>
</tr>
</tbody>
</table>

Table 2.1: Geometric errors for lines in Fig. 2.14 where affine models intersect. Errors are computed as the sum of distances between the ground truth line segment corners and the computed lines. The errors for sequential RANSAC and J-linkage, see Fig. 2.15, were significantly larger than those above.

we manually extracted the lines corresponding to intersecting planes. Assuming that two intersecting planes $\pi_1$ and $\pi_2$ are represented by the affine models $A^{\pi_1}$ and $A^{\pi_2}$, the homogeneous vector representing the 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 BT or PEARL. Table 2.1 compares the accuracy of these lines with respect to our ground truth.

2.3.2 Estimating Multiple Homographies

In this section we use PEARL to estimate multiple homographies in uncalibrated wide-base stereo image pairs. We use SIFT features [20] as points of interest. The set of all matched pairs of features $P$ is found using exhaustive search.

One way to measure geometric error between a pair of points $p = (p_l, p_r)$ and a given homography $H$ is to use the symmetric transfer error (STE) [16]. We generate our finite set of initial model proposals by randomly sampling four matched pairs at a time. Model parameters are computed by minimizing the non-linear STE error using Levenberg-Marquardt, as described in [16]. The initial solution for the non-linear minimization is
Figure 2.14: Comparison of results by BT [2] and PEARL. Lines are computed for all pairs of intersecting planes (affine models).

Figure 2.15: Typical results for affine model fitting in Sec. 2.3.1 via sequential RANSAC (a) and J-linkage [26]. We used 5000 uniform samples for RANSAC. Different threshold values did not give much improvement. Similarly, different tunings of J-linkage generated various artifacts. These problems might be explained by the large level of noise in the data, as in Fig. 2.9.
2.3. Experimental Results

Figure 2.16: Multi-homography fitting, images from VGG (Oxford) Merton College I.

found using the *direct linear transform* (DLT) method. We apply PEARL to energy

$$E(H, f) = \sum_{p \in P} ||p - Hf_p|| + \lambda \sum_{(p, q) \in N} [f_p \neq f_q] + \beta \cdot \sum_{h \in \mathcal{L}} \delta_h(f)$$  \hspace{1cm} (2.11)

where $H = \{H_h \mid h \in \mathcal{L}\}$ is a set of homographies and the neighborhood system $N$ is based on a triangulation of data points in one of the images.

In the example of Fig. 2.16(a) PEARL identified 7 planes. The third term in energy (2.11) allowed the merging of spatially isolated parts of yellow, green, and white planes. Unlike multi-RANSAC [40], PEARL does not require *a priori* knowledge of the number of planes and produces spatially coherent inliers. In [40] multi-RANSAC required 11604 iterations to fit 4 models to the same data, see Fig. 2.16(b). Since each iteration sampled 4 random models, the total number of sampled homographies in Fig. 2.16(b) is 46416. In contrast, PEARL used only 900 randomly sampled models to identify 7 planes. PEARL converged in three iterations. For qualitative comparison, Fig. 2.16 also shows the result based on spectral clustering from [6] and the best result we could obtain using J-linkage [26].

In Fig. 2.17(a) PEARL identified 8 planes using only 3000 initial samples and con-
Figure 2.17: Multi-homography fitting for stereo images from VGG (Oxford) Merton College III. Both results were obtained using standard SIFT features.

Figure 2.18: Multi-homography fitting for “stairs” stereo image from VGG (Oxford). The results in (b) and (c) represent our best efforts in tuning RANSAC and J-linkage thresholds. For example, two “walls” start to grab false positive matches for larger threshold values, while for smaller thresholds the “stairs” become even more over-segmented compared to what is shown here.
2.3. Experimental Results

Figure 2.19: In this Raglan Castle example from VGG (Oxford) we used the same color more than once to represent different planes (a). Only spatially connected planes are shown in different colors. An image of the same scene from a different view (b) confirms that the vertical walls of the building corresponds to different planes.
Ch. 2. Energy-based Geometric Multi-Model Fitting

verged after four iterations. PEaRL finds planes with varying number of inliers. The roof on the left (light green) has only 13 inliers, while the two large walls (blue and pink) have 786 and 581 inliers, respectively. The qualitative comparison of our regularization-based method with greedy techniques like J-Linkage and sequential RANSAC in Figures 2.17 and 2.18 shows that they are less robust to high noise. Similar general limitation of greedy inlier maximization approaches was previously demonstrated on synthetic data in Figures 2.1 and 2.9.

Figure 2.19 shows Raglan Castle Tower result for PEaRL using 6000 initial labels and only four iterations to convergence. PEaRL identified 13 planes. The use of a relatively large number of initial labels allowed PEaRL to identify very small planes. Another picture of Raglan Castle Tower from flicker confirms that the walls on the second and the third floors represent different parallel planes.

2.3.3 Motion Segmentation

In this section we aim to solve the multi-body motion segmentation problem using multiple-views. This problem is also referred to in literature as the multibody structure from motion problem [4, 27, 10]. The goal of this problem is to cluster the scene trackable features among distinct motions, then to estimate the motions’ parameters and to recover the 3D structure of the points. We are only interested in estimating the multiple motions and clustering of image features.

Using two-views

Assume that the multiple bodies are rigid and each body undergoes a different motion. Each distinct rigid-body motion with rotation $R$ and translation $T$ could be described by a fundamental matrix $F = [K’]_r K’ R K^{-1}$ corresponding to two views where $K$ and $K’$ are the calibration matrices of the used cameras. This fundamental matrix satisfies the epipolar constraint $p_r^T F p_l = 0$ where $p_r$ and $p_l$ are two matching features corresponding to a 3D point on some rigid body [21, 19].

We apply PEaRL to estimate multiple fundamental matrices for uncalibrated image pairs. Matching pairs are found using the same procedure mentioned in section 2.3.2. One way to measure geometric error between a matching pair of points $p$ and a given model $F$ is the Squared Sampson Distance (SSD) [16]

$$||p - F|| = \frac{(p_r^T F p_l)^2}{(F p_l)^2 + (F p_r)^2 + (F^T p_r)^2 + (F^T p_r)^2}$$

(2.12)

where the $(F p_l)^2$ represents the square of the $j^{th}$ entry of the vector $F p_l$. We generate our finite set of initial model proposals by random sampling eight matching pairs at a time. Then compute the model parameters as described in [16] by minimizing the non-linear SSD error using Levenberg-Marquardt. The initial solution for the non-linear minimization is found using the normalized 8-point algorithm [17]. The next step is to triangulate the features of one of the images (e.g. the right image). Then we apply PEaRL to energy

$$E(F, f) = \sum_{p \in P} ||p - F p|| + \lambda \sum_{(p, q) \in N} [f_p \neq f_q] + \beta \sum_{h \in L} \delta_h (f)$$

(2.13)
### 2.3. Experimental Results

<table>
<thead>
<tr>
<th># frames used</th>
<th>PEARL</th>
<th>ER1</th>
<th>ER2</th>
<th>REF</th>
<th>GPCA</th>
<th>LSA 4n</th>
<th>RANSAC</th>
</tr>
</thead>
<tbody>
<tr>
<td># models fixed</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>average</td>
<td>21.2%</td>
<td>6.93%</td>
<td>4.2%</td>
<td>6.28%</td>
<td>31.95%</td>
<td>5.80%</td>
<td>25.78%</td>
</tr>
<tr>
<td>median</td>
<td>11.9%</td>
<td>4.15%</td>
<td>1.2%</td>
<td>5.06%</td>
<td>32.93%</td>
<td>1.77%</td>
<td>26.01%</td>
</tr>
</tbody>
</table>

Table 2.2: Misclassification errors for checkerboard motion estimation sequences in [31]. The results for REF, GPCA [32], LSA [36], and RANSAC are copied from Table 4 in [31]. These methods use all N-frames in each video sequence. In contrast, our results for PEARL with energy (2.13) use only 2 frames, the first and the last in the sequence. Other methods also “know” that the exact number of models is 3. In contrast, PEARL may obtain a different number of models. Figure 2.22 illustrates how this affects PEARL’s classification errors. For a more balanced comparison we show 2 additional statistics. First, column ER1 evaluates 15 out of 26 examples where PEARL obtained exactly 3 models. Second, similarly to REF [31], column ER2 evaluates the “reference” solutions obtained by PEARL from the ground truth models in all 26 examples.

where $\mathbf{F} = \{F_h \mid h \in \mathcal{L}\}$ is the a set of fundamental matrices. Figure 2.20 shows qualitative results. In case the label cost term in (2.13) is dropped, it is likely that spatially isolated parts of the background could be assigned different motions, as shown in Fig. 2.21.

Table 2.2 compares our results with some standard motion estimation methods evaluated in [31]. Note that these standard methods assume that the number of motions in the data is given. In contrast, our method automatically estimates the number of motions. The comparison in Table 2.2 may not be entirely meaningful since other methods benefit from a priori knowledge of the exact number of motions, while our method may get an incorrect number of motions contributing to gross misclassification errors, see Fig. 2.22. One way to alleviate this problem is to report PEARL’s statistics on examples where the number of models was estimated correctly, see column ER1. We also evaluated the “reference” solutions obtained when PEARL is initialized with 3 ground truth models, see column ER2.

Statistically the required number of samples needed to generate good samples of fundamental matrices in the motion examples could be quite large. For example, if data points contain only 3 motions with $m_i$ inliers (each) and $m_o$ outliers, one can compute the probability that $n$ independent samples have at least one good representative of each model

$$
\Pr(3 \text{ out of } n) = 1 - (1 - \rho_1)^n - (1 - \rho_2)^n - (1 - \rho_3)^n
$$

$$
+ (1 - \rho_1 - \rho_2)^n + (1 - \rho_2 - \rho_3)^n + (1 - \rho_3 - \rho_1)^n
$$

$$
- (1 - \rho_1 - \rho_2 - \rho_3)^n
$$

where $\rho_i \approx \left( \frac{m_i}{m_1 + m_2 + m_3 + m_o} \right)^k$ is the probability that $k$ randomly selected points (for estimating a model) come exclusively from $m_i$ inliers of the $i$th model. In checkerboard sequences [31] there are 3 models with $m_1 = 20$, $m_2 = 24$, and $m_3 = 56$ inliers and
Figure 2.20: Representative PEaRL’s results for motion sequences from Rene Vidal’s data set [31]. Our motion estimation using mixtures of fundamental matrices (Sec. 2.3.3) uses only two frames (the first and the last).
2.3. Experimental Results

Figure 2.21: Optimal solution for energy (2.3). In contrast to results for energy (2.7) in Fig.2.20 (e)-(h), it fails to generate one background motion from yellow, green, and red clusters. These clusters correspond to infinitesimally close motions, but they are spatially isolated. The third term in (2.7) addresses this issue.

Figure 2.22: Histogram of misclassification errors for PEARL with energy (2.13) on 26 Checkerboard examples [31] with 3 distinct motions. This histogram reveals two “modes”. The mode with smaller misclassification errors mainly contains examples where PEARL produced exactly 3 models. Such examples are marked in blue. Examples where PEARL produced a different number of models are marked in red. Such examples formed the “gross errors” mode. The percentage of misclassified points may not be a proper measure for comparing a method that automatically computes the number of models against the methods that a priori know the correct number. We separately report PEARL’s error statistics for 15 (blue) examples with the correct number of obtained models, see ER1 in Table 2.2.
Figure 2.23: Scatter plot of different solutions obtained by PEARL on one of the examples in [31] using different sets of 5000 (uniformly) sampled fundamental matrices. This number of samples is statistically insufficient for the motion estimation examples in Sec. 2.3.3 (see text) and the quality of optimization may suffer. The plot also illustrates positive correlation between the values of our energy (2.13) and the misclassification errors.

The results in this section were obtained by sampling the data points uniformly. Instead, we can use any standard non-uniform local sampling scheme (see [7] for a recent review). The main effect of using such sampling schemes in the propose step of PEARL could be lowering the number of samples required to find a good (low-energy) solution. A detailed study of this effect is outside the scope of this work.

Note that PEARL's results in Table 2.2 are obtained assuming the same level of noise for all models. However, adding noise level as a parameter for each model, see Fig. 2.11, could enhance our results.
Using multiple views/frames

Our general energy-based model fitting approach can also be applied to the N-frame point-trajectory data used by the motion estimation methods evaluated in [31]. In fact, the standard N-frame methodology introduced by Tomasi and Kanade [27] may lead to a more accurate and faster technique. For example, instead of fundamental matrices with 7 degrees of freedom, each rigid motion is represented as a lower dimensional (4D or less) linear manifold in the space of motion trajectories. This representation assumes affine projection.

There are many ways in which our general optimization-based model fitting framework can be applied in this context. For example, it may be used as a trajectory grouping technique instead of spectral clustering in GPCA [32] or LSA [36]. However, given the general scope of this paper, we present only the most straightforward set-up for fitting 4D hyperplanes following the basic formulation used in [31] for sequential RANSAC.

In contrast to greedy maximization of inliers by RANSAC, we seek an optimal labeling of data points. As in [28, 25], our general energy optimization framework can assign different label costs for fitting motions of various complexities, e.g. degenerate and non-degenerate. We can also replace piece-wise constant spatial regularization by piece-wise smooth in order to better address non-independent or articulated motions. For simplicity, however, we stick to the most basic formulation as described below.

Assuming that the cameras are affine then it could be proven that the motion of a rigid body i.e. the trajectory of its features will lie in a 4D subspace [4, 27, 32]. Let \( P \) be a set of 3D points that belong to a rigid body which undergoes a motion over \( N \) frames. The image feature of a point \( i \) in frame \( k \) is defined by \( p_{ki} \). Stacking all \( p_{ki} \) measurements \( \forall k \in N \) and \( \forall i \in P \) will form the measurement matrix

\[
W = \begin{bmatrix}
p_{11} & \cdots & p_{1P} \\
\vdots & & \vdots \\
p_{N1} & \cdots & p_{NP}
\end{bmatrix}_{2N \times P} \tag{2.14}
\]

which has rank 4 [4, 27]. First, we project these feature trajectories from \( \mathbb{R}^{2N} \) to \( \mathbb{R}^5 \). The extra dimension is needed to discriminate between different motions which will be hyperplanes in \( \mathbb{R}^5 \) [32]. Then we use PEARL to fit multiple 4D hyperplanes (motions). In contrast to [32, 36], PEARL does not require prior knowledge of the number of motions.

We generate a finite set of initial proposals by randomly sampling four points, i.e. four projected trajectories and the fitting a 4D hyperplane to it. The next step is to triangulate the 2D image features on one of the frames, e.g. the last frame. Finally, we apply PEARL to energy

\[
E(M, f) = \sum_{p \in P} ||p - M_{fp}|| + \lambda \sum_{(p,q) \in N} [f_p \neq f_q] + \beta \sum_{h \in L} \delta_h(f) \tag{2.15}
\]

where \( M = \{M_h \mid h \in L\} \) is a set of 4D hyperplane. Figure 2.24 shows some qualitative results obtained by PEARL for energy (2.15) using the same set of parameters. Figure 2.25 indicates how the classification errors may depend on the main parameters in (2.15): weight of the smoothness term \( \lambda \) and weight of the label cost term \( \beta \).
Figure 2.24: Representative PEARL’s results using energy (2.15) for motion sequences from Vidal’s data set [31]. Unlike the fundamental matrix fitting method in Sec. 2.3.3, this formulation uses all frames in the sequence.
2.4 Conclusions

We proposed a new general approach to geometric multi-model fitting based on global optimization. The problem is formulated as discrete labeling of data points using MRF and MDL style regularization functionals widely used in other computer vision problems. The goal is to find models “explaining” all data points based on spatial regularity and sparsity priors. The continuous space of model parameters is explored via PEaRL algo-
Table 2.3: Misclassification errors for checkerboard motion estimation sequences in [31]. The results for REF, GPCA [32], LSA [36], and RANSAC are copied from Table 4 in [31]. All methods in this table use all frames in each video sequence. PEARL may obtain a different number of models while the other methods “know” that the exact number of models is 3. For a more balanced comparison we show 2 additional statistics. First, column ER1 evaluates 12 out of 26 examples where PEARL obtained exactly 3 models. Second, similarly to REF in [31], column ER2 evaluates the “reference” solutions obtained by PEARL from the ground truth models in all 26 examples.

The algorithm that combines data sampling and energy minimization iterating assignment and re-estimation steps. The method automatically obtains a small number of models that “explain” data well. Many empirical tests on synthetic and real imagery demonstrate a strong potential of our general approach applicable to a wide spectrum of model fitting problems. The main conclusion are:

- The proposed general energy functionals (2.3) and (2.7) are potent criteria for solving a wide class of geometric multi-model fitting problems with a priori unknown number of models corrupted by noise and outliers.

- The proposed algorithmic approach for minimizing our model fitting energies (PEARL) works well in many practical applications.

The two main parameters in the proposed energies (2.3) and (2.7) are the weights \( \lambda \) and \( \beta \) for the spacial regularity and label cost terms. We did not analyze any specific technique for selecting these parameters, but we demonstrate that the method is fairly stable with respect to them. Many useful ideas for parameter selection can be borrowed from information theoretic interpretation of these energies, e.g. see [28, 11].

One can use any geometric error function \( ||p - \theta_h|| \) in the data term to evaluate the distance between models and data points. We also showed that fitting one additional uniform error model \( \phi \) may work well for classifying the outliers. Our general energy formulation does not require any hard thresholds, even though one can use truncated or step error measures \( ||p - \theta_h|| \), if necessary.
Bibliography


Chapter 2. Energy-based Geometric Multi-Model Fitting


Chapter 3

Fast Approximate Energy Minimization with Label Costs

The \( \alpha \)-expansion algorithm [4] has had a significant impact in computer vision due to its generality, effectiveness, and speed. Thus far it can only minimize energies that involve unary, pairwise, and specialized higher-order terms. Our main contribution is to extend \( \alpha \)-expansion so that it can simultaneously optimize “label costs” as well. An energy with label costs can penalize a solution based on the set of labels that appear in it. The simplest special case is to penalize the number of labels in the solution. Our energy is quite general, and we prove optimality bounds for our algorithm. A natural application of label costs is multi-model fitting, and we demonstrate several such applications in vision: homography detection, motion segmentation, and unsupervised image segmentation. Also, the C++ implementation is publicly available.

3.1 Some Useful Regularization Energies

In a labeling problem we are given a set of observations \( \mathcal{P} \) (pixels, features, data points) and a set of labels \( \mathcal{L} \) (categories, geometric models, disparities). The goal is to assign each observation \( p \in \mathcal{P} \) a label \( f_p \in \mathcal{L} \) such that the joint labeling \( f \) minimizes some objective function \( E(f) \).

Most labeling problems in computer vision are ill-posed and in need of regularization, but the most useful regularizers often make the problem NP-hard. Our work is about how to effectively optimize two such regularizers: a preference for fewer labels in the solution, and a preference for spatial smoothness. Figure 3.1 suggests how these criteria cooperate to give clean results. Surprisingly, there is no good algorithm to optimize their combination. Our main contribution is a way to simultaneously optimize both of these criteria inside the powerful \( \alpha \)-expansion algorithm [4].

Label costs: Start from a basic (unregularized) energy \( E(f) = \sum_{p} D_p(f_p) \), where the optimal \( f_p \) can each be determined independently from the ‘data costs’. Suppose, however, that we wish to explain the observations using as few unique labels as necessary. We can introduce label costs into \( E(f) \) to penalize each unique label that appears in \( f \):

\[
E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{l \in \mathcal{L}} h_l \cdot \delta_l(f) \tag{3.1}
\]
3.1. Some Useful Regularization Energies

Figure 3.1: Motion segmentation on the 1RT2RCR sequence [25]. Energy (3.1) finds 3 dominant motions (b) but labels many points incorrectly. Energy (3.2) gives coherent segmentations (c) but finds redundant motions. Our energy (⋆) combines the best of both (d).
where \( h_l \) is the non-negative label cost of label \( l \), and \( \delta_l(\cdot) \) is the corresponding indicator function

\[
\delta_l(f) \overset{\text{def}}{=} \begin{cases} 
1 & \exists p : f_p = l \\
0 & \text{otherwise.}
\end{cases}
\]

Energy (3.1) balances data costs against label costs in a formulation equivalent to the well-studied uncapacitated facility location (UFL) problem. Li [18] recently posed multi-model fitting in terms of UFL. For multi-model fitting, where each label corresponds to a candidate model, label costs penalize overly-complex models, preferring to explain the data with fewer, cheaper labels (see Figure 3.1a).

**Smooth costs:** Spatial smoothness is a standard regularizer in computer vision. The idea here is that groups of observations are often known a priori to be positively correlated, and should thus be encouraged to have similar labels. Neighbouring image pixels are a classic example of this. Such pairwise priors can be expressed by the energy

\[
E(f) = \sum_{p \in P} D_p(f_p) + \sum_{pq \in N} V_{pq}(f_p, f_q)
\]

where each \( V_{pq} \) penalizes \( f_p \neq f_q \) in some manner. If each \( V_{pq} \) defines a metric, then minimizing (3.2) is known as the metric labeling problem [4] and can be optimized effectively with the \( \alpha \)-expansion algorithm.

This regularizer prefers coherent segmentations, but has no incentive to combine non-adjacent segments and thus a tendency to suggest redundant labels in multi-model fitting (see Figure 3.1b). Still, spatial smoothness priors are important for a wide array of vision applications.

**Our combined energy:** We propose a discrete energy that essentially combines the UFL and metric labeling problems.

\[
E(f) = \sum_{p \in P} D_p(f_p) + \sum_{pq \in N} V_{pq}(f_p, f_q) + \sum_{L \subseteq L} h_L \cdot \delta_L(f) \quad (\star)
\]

where the indicator function \( \delta_L(\cdot) \) is now defined on label subset \( L \) as

\[
\delta_L(f) \overset{\text{def}}{=} \begin{cases} 
1 & \exists p : f_p \in L \\
0 & \text{otherwise.}
\end{cases}
\]

Our energy actually makes a slight generalization from label costs to label subset costs \( h_L \), but one can imagine simple per-label costs \( h_l \) throughout for simplicity.

Energy (\( \star \)) balances two demonstrably important regularizers, as suggested by Figure 3.1c. Figures 3.2 and 3.3 show other vision applications where our combined energy simply makes sense. Section 3.2 presents our extension to \( \alpha \)-expansion and corresponding optimality bounds. Sections 3.3 and 3.4 explain how our energy can be effective as a multi-model fitting framework. See Section 3.5 for discussion and possible extensions, and see [8] for relation to standard expectation maximization (EM) and K-means formulations.
3.1. Some Useful Regularization Energies

Figure 3.2: Planar homography detection on VGG (Oxford) Merton College 1 image (right view). Energy (3.1) finds reasonable parameters for only the strongest 3 models shown in (b), and still assigns a few incorrect labels. Energy (3.2) finds reasonable clusters (c) but fits 9 models, some of which are redundant (nearly co-planar). Our energy (∗) finds both good parameters and labels (d) for 7 models.
Figure 3.3: Unsupervised segmentation using histogram models. Energy (3.1) clusters in
colour space, so segments (b) are incoherent. Energy (3.2) clusters over pixels and must
either over-segment or over-smooth (c), just as in [26]. Our energy (⋆) balances these
criteria (d) and corresponds to Zhu & Yuille [27] for segmentation.
3.2 Fast Algorithms to Minimize (⋆)

Our main technical contribution is to extend the well-known \(\alpha\)-expansion algorithm \([4]\) to incorporate label costs at each expansion (Section 3.2.1) and prove new optimality guarantees (Section 3.2.2). Section 3.2.3 reviews known results for the ‘easy’ case (3.1) with only data and per-label costs.

### 3.2.1 Expansion moves with label costs

Since minimizing energy (⋆) is NP-hard for \(|L| \geq 3\), the \(\alpha\)-expansion algorithm \([4]\) iteratively ‘moves’ from some current labeling \(f'\) to a better one until convergence. Specifically, at each step, some label \(\alpha \in L\) is chosen and variables \(f_p\) are simultaneously given a binary choice to either stay as \(f_p = f'_p\) or switch to \(f_p = \alpha\). This binary step is called expansion because only the \(\alpha\) label can grow and, if each \(V_{pq}\) satisfies a simple condition, the best possible expansion is computed by a single graph cut.

Let \(f = \{f_1, \ldots, f_n\}\) and let \(f^\alpha\) denote any feasible \(\alpha\)-expansion w.r.t. current labeling \(f'\). The possible labelings \(f^\alpha\) can be expressed one-to-one with binary indicator variables \(x = \{x_1, \ldots, x_n\}\) by defining

\[
\begin{align*}
x_p &= 0 \iff f^\alpha_p = f'_p \\
x_p &= 1 \iff f^\alpha_p = \alpha.
\end{align*}
\]  

(3.3)

Let \(E^\alpha(x)\) be the energy corresponding to encoding (3.3) relative to \(f'\). The \(\alpha\)-expansion algorithm computes an optimum \(x^*\), and thereby \(f^\alpha\), by a single graph cut.

For example, suppose energy \(E(f)\) is such that the optimal expansion w.r.t labeling \(f'\) is \(f^\alpha\):

\[
f' = \begin{bmatrix} \beta & \alpha & \gamma & \gamma & \beta & \beta \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} \alpha & \alpha & \alpha & \gamma & \beta & \beta \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} = f^\alpha
\]

(3.4)

where 1 means \(x_2\) is fixed to 1. Here only \(f_1\) and \(f_5\) changed to label \(\alpha\) while the rest preferred to keep their labels. The \(\alpha\)-expansion algorithm iterates the above binary step until finally \(E^\alpha(x') = E^\alpha(x^*)\) for all \(\alpha \in L\).

**Encoding label costs.** The energy in example (3.4) was such that \(f_5\) and \(f_6\) preferred to stay as label \(\beta\) rather than switch to \(\alpha\). Suppose we want to introduce a cost \(h_\beta > 0\) that is added to \(E(f)\) if and only if there exists some \(f_p = \beta\). This would encourage label \(\alpha\) to absorb the entire region that \(\beta\) occupies in \(f'\). If \(h_\beta\) is large enough, the optimal \(\alpha\)-expansion move would also change \(f_5\) and \(f_6\):

\[
f' = \begin{bmatrix} \beta & \alpha & \gamma & \gamma & \beta & \beta \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} \alpha & \alpha & \alpha & \gamma & \beta & \beta \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} = x^*
\]

(3.5)

Our main algorithmic contribution is a way to encode such label costs into the expansion step and thereby encourage solutions that use fewer labels.

Energy \(E^\alpha(x)\), when expressed as a multilinear polynomial, is a sum of linear and quadratic terms over \(x\). For the specific example (3.5), we can encode cost \(h_\beta\) in \(E^\alpha\) by...
simply adding $h_{\beta} - h_{\beta}x_1x_5x_6$ to the binary energy. Because this specific term is cubic and $h_{\beta} \geq 0$, it can be optimized by a single graph cut using the construction in [15].

To encode general label costs for arbitrary $L \subseteq \mathcal{L}$ and $f'$, we must optimize the modified expansion energy

$$E_{h}^\alpha(x) = E^\alpha(x) + \sum_{L \subseteq \mathcal{L} \backslash \mathcal{L}' \neq \emptyset} \left( h_L - h_L \prod_{p \in \mathcal{P}_L} x_p \right) + C^\alpha(x) \quad (3.6)$$

where set $\mathcal{L}'$ contains the unique labels in the current labeling $f'$, and set $\mathcal{P}_L = \{ p : f'_p \in L \}$. Term $C^\alpha$ simply corrects for the case when $\alpha \notin \mathcal{L}'$ and is discussed later.

Each product term in (3.6) adds a higher-order clique $\mathcal{P}_L$ beyond the standard $\alpha$-expansion energy $E^\alpha(x)$. Freedman and Drineas [10] generalized the graph construction of [15] to handle terms $c \prod_p x_p$ of arbitrary degree when $c \leq 0$. This means we can transform each product seen in (3.6) into a sum of quadratic and linear terms that graph cuts can still optimize globally. The transformation for a particular label subset $L \subseteq \mathcal{L}$ with $|\mathcal{P}_L| \geq 3$ is

$$-h_L \prod_{p \in \mathcal{P}_L} x_p = \min_{y_L \in \{0,1\}} h_L \left[ (|\mathcal{P}_L|-1)y_L - \sum_{p \in \mathcal{P}_L} x_p y_L \right] \quad (3.7)$$

where $y_L$ is an auxiliary variable that, if $h_L > 0$, must be optimized alongside $x$. Since each $x_p y_L$ term has non-positive coefficient, it can be optimized by graph cuts.

To encode the potential (3.7) into an $s$-$t$ min-cut graph construction, we re-parameterize the right-hand side such that each quadratic monomial has exactly one complemented variable (e.g. $x \bar{y}$) and non-negative coefficient (arc weight). One such re-parameterization is

$$-h_L + h_L \bar{y}_L \sum_{p \in \mathcal{P}_L} h_{\bar{L}} \bar{x}_p y_L \quad (3.8)$$

where $\bar{x}_p = 1 - x_p$. Figure 3.4 shows the subgraph corresponding to (3.8) after canceling the constant $-h_L$. 

![Graph construction](image-url)
Subgraphs of this type have been used in vision before, most notably the $P^n$ Potts potentials of Kohli et al. [13]. Our indicator potentials $\delta_L(\cdot)$ are different in that, at the binary step (3.6), each clique $P_L$ is determined \textit{dynamically} from the current labeling $f'$ and is not expressed as such in the original energy $(\ast)$. It is easy to represent a $P^n$ Potts potential by a combination of label subset cost potentials, but not the other way around. Our technical report [8] elaborates on this point and Section 3.5 mentions an extension to our work based on the Robust $P^n$ Potts construction [14].

A final detail is how to handle the case when $\alpha$ was not used in the current labeling $f'$. The corrective term $C^\alpha$ in (3.6) incorporates the label costs for $\alpha$ itself:

$$C^\alpha(x) = \sum_{L \subseteq \mathcal{L}} \left( h_L - h_L \prod_{p \in P} \bar{x}_p \right).$$

(3.9)

If we find that $x^* = 0$ then label $\alpha$ was not used in $f'$ and it was also not worth expanding it in $f^\alpha$. The term (3.9) can be encoded by a subgraph analogous to Figure 3.4, but the following is more efficient: first compute an optimal $x^*$ for (3.6) without considering $C^\alpha$, then explicitly add it to $E^\alpha_h(x^*)$ if $x^* \neq 0$, and reject the expansion if the energy would increase. In fact, a similar test-and-reject step allows label costs to be trivially incorporated into $\alpha$-$\beta$-swap: before accepting a swap move, test its energy against the energy when all $\beta$ variables become $\alpha$ and vice versa.

### 3.2.2 Optimality guarantees

In what follows we assume that energy $(\ast)$ is configured\footnote{Adding an arbitrary constant to $D_p(\cdot)$ or $V_{pq}(\cdot, \cdot)$ does not affect the optimal labeling, so finite costs can always be made non-negative.} so that $D_p \geq 0$, $V_{pq}$ is a metric [4], and thus $E(f) \geq 0$.

**Theorem 1** If $f^*$ is a global optimum of energy $(\ast)$ and $\hat{f}$ is a local optimum w.r.t. $\alpha$-expansion then

$$E(\hat{f}) \leq 2cE(f^*) + \sum_{L \subseteq \mathcal{L}} h_L |L|$$

(3.10)

where

$$c = \max_{pq \in \mathcal{N}} \left( \max_{\alpha \neq \beta \in \mathcal{C}} V_{pq}(\alpha, \beta) / \min_{\gamma \neq \zeta \in \mathcal{C}} V_{pq}(\gamma, \zeta) \right)$$

**Proof of Theorem 1.** The proof idea follows Theorem 6.1 of [4]. Let us fix some $\alpha \in \mathcal{L}$ and let

$$\mathcal{P}_\alpha \overset{\text{def}}{=} \{ p \in \mathcal{P} : f^*_p = \alpha \}.$$  

(3.11)

We can produce a labeling $f^\alpha$ within one $\alpha$-expansion move from $\hat{f}$ as follows:

$$f^\alpha_p = \begin{cases} \alpha & \text{if } p \in \mathcal{P}_\alpha \\ f^*_p & \text{otherwise.} \end{cases}$$

(3.12)
Since \( \hat{f} \) is a local optimum w.r.t. expansion moves we have
\[
E(\hat{f}) \leq E(f^\circ). \tag{3.13}
\]

Let \( E(\cdot)|_S \) denote a restriction of the summands of energy (\(*\)) to only the following terms:
\[
E(f)|_S = \sum_{p \in S} D_p(f_p) + \sum_{pq \in S} V_{pq}(f_p, f_q). 
\]

We separate the unary and pairwise terms of \( E(f) \) via interior, exterior, and boundary sets with respect to pixels \( P_\alpha \):
\[
I_\alpha = P_\alpha \cup \{pq \in N: p \in P_\alpha, q \in P_\alpha\} \\
O_\alpha = P \setminus P_\alpha \cup \{pq \in N: p \notin P_\alpha, q \notin P_\alpha\} \\
B_\alpha = \{pq \in N: p \in P_\alpha, q \notin P_\alpha\}.
\]

The following facts now hold:
\[
E(f^\circ)|_{I_\alpha} = E(f^*)|_{I_\alpha} \tag{3.14} \\
E(f^\circ)|_{O_\alpha} = E(\hat{f})|_{O_\alpha} \tag{3.15} \\
E(f^\circ)|_{B^\alpha} \leq cE(f^*)|_{B^\alpha}. \tag{3.16}
\]

Equation (3.16) holds because for any \( pq \in B_\alpha \) we have \( V_{pq}(f^\circ_p, f^\circ_q) \leq cV_{pq}(f^*_p, f^*_q) \).

Let \( E_H \) denote the label cost terms of energy \( E \). Using (3.14), (3.15) and (3.16) we can rewrite (3.13) as
\[
E(\hat{f})|_{I_\alpha} + E(\hat{f})|_{B^\alpha} + E_H(\hat{f}) \tag{3.17} \\
\leq E(f^\circ)|_{I_\alpha} + E(f^\circ)|_{B^\alpha} + E_H(f^\circ) \tag{3.18} \\
\leq E(f^*)|_{I_\alpha} + cE(f^*)|_{B^\alpha} + E_H(f^\circ) \tag{3.19}
\]

Depending on \( \hat{f} \) we can bound \( E_H(f^\circ) \) by
\[
E_H(f^\circ) \leq E_H(\hat{f}) + \left\{ \begin{array}{ll}
h_L & \text{if } \alpha \in \mathcal{L}^* \\
\sum_{\substack{\mathcal{L} \subseteq \mathcal{L} \setminus \hat{\mathcal{L}}} \mathcal{L} \cap \mathcal{L}_\alpha} h_L & \text{otherwise.} \\
\end{array} \right. \tag{3.20}
\]

where sets \( \mathcal{L}^* \) and \( \hat{\mathcal{L}} \) contain the unique labels in \( f^* \) and \( \hat{f} \) respectively.

To bound the total energy we sum expressions (3.17) and (3.19) over all labels \( \alpha \in \mathcal{L}^* \) to arrive at the following:
\[
\sum_{\alpha \in \mathcal{L}^*} \left( E(\hat{f})|_{I_\alpha} + E(\hat{f})|_{B^\alpha} \right) \tag{3.21} \\
\leq \sum_{\alpha \in \mathcal{L}^*} \left( E(f^*)|_{I_\alpha} + cE(f^*)|_{B^\alpha} \right) + \sum_{\mathcal{L} \subseteq \mathcal{L} \setminus \hat{\mathcal{L}}} h_L |\mathcal{L} \cap \mathcal{L}^*|.
\]
Observe that, for every \( pq \in \mathcal{B} = \bigcup_{\alpha \in \mathcal{L}} \mathcal{B}_\alpha \), the term \( V_{pq}(\hat{f}_p, \hat{f}_q) \) appears twice on the left side of (3.21), once for \( \alpha = f_p^* \) and once for \( \alpha = f_q^* \). Similarly every \( V(f_p^*, f_q^*) \) appears \( 2c \) times on the right side of (3.21). Therefore equation (3.21) can be rewritten as

\[
E(\hat{f}) \leq E(f^*) + (2c - 1)E(f^*)|_\mathcal{B} - E(\hat{f})|_\mathcal{B} + E_H(\hat{f}) - E_H(f^*) + \sum_{L \subseteq \mathcal{L} \setminus \hat{\mathcal{L}}} h_L|L \cap \mathcal{L}^*|.
\]

The above inequality is a tight \textit{a posteriori} bound on \( E(\hat{f}) \) w.r.t. a specific local optimum \( \hat{f} \) and global optimum \( f^* \); see [8] for worst-case local minima. Observe that

\[
E_H(\hat{f}) - E_H(f^*) + \sum_{L \subseteq \mathcal{L} \setminus \hat{\mathcal{L}}} h_L|L \cap \mathcal{L}^*| = \sum_{L \subseteq \mathcal{L} \setminus \mathcal{L}^*} h_L + \sum_{L \subseteq \mathcal{L} \setminus \hat{\mathcal{L}}} h_L(|L \cap \mathcal{L}^*| - 1) \\
\leq \sum_{L \subseteq \mathcal{L}} h_L|L|.
\]

Using (3.23) and the assumption \( D_p \geq 0 \) we simplify (3.22) to give \textit{a priori} bound (3.10).

The \textit{a priori} bound (3.10) suggests that for large label costs the worst-case approximation is poor. The fundamental problem is that \( \alpha \)-expansion can expand only one label at a time. It may help empirically to order the expansions in a greedy manner, but the next section describes a special case for which the greedy algorithm still yields a similar additive bound (see Section 3.5.1 of [7]). We thus do not expect much improvement unless different moves are considered.

### 3.2.3 Easy case: only per-label costs

In the absence of any smooth costs (\( V_{pq} \equiv 0 \)) and higher-order label costs (\( h_L = 0 \) for \( |L| > 1 \)) our energy reduces to a special case (3.1) known as the \textit{uncapacitated facility location} (UFL) problem. This well-studied problem was recently applied for motion segmentation, first by Li [18] and then by Lazic et al. [17]. The UFL problem assigns facilities (labels) to each client (variable) such that the cost to clients is balanced against the cost of ‘opening’ facilities to serve them. Optimizing UFL is NP-hard by simple reduction from \textit{Set-Cover}, so it is ‘easier’ than our full energy (\( \ast \)) only in a practical sense.

Li optimizes the integer program corresponding to UFL by \textit{linear programming (LP) relaxation}, then rounds fractional ‘facility’ variables to 0 or 1 in a straight-forward manner. Because of the heavy LP machinery, this approach is slow and affords relatively few candidate models in practice. Li implements \textit{four} application-specific heuristics to aggressively prune candidate models “for LP’s sake.” Lazic et al. optimize the same energy using max-product belief propagation (BP), a message-passing algorithm.
Kuehn & Hamburger [16] proposed a natural greedy algorithm for UFL in 1963. The algorithm starts from an empty set of facilities (labels) and greedily introduces one facility at a time until no facility would decrease the overall cost. The greedy algorithm runs in $O(|L|^2|P|)$ time for labels $L$ and observations $P$. Hochbaum [12] later showed that the greedy algorithm yields a $O(\log |P|)$-approximation in general, and better bounds exist for special cost functions (see [22] for review). Other greedy moves have been proposed for UFL besides “open one facility at a time” (see [6, 7]).

The C++ library\(^2\) implements the greedy heuristic [16] and, when smooth costs are all zero, it is 15–30 times faster than $\alpha$-expansion while yielding similar energies. Indeed, “open facility $\alpha$” is equivalent to expansion in this case. Note that our higher-order label costs can also be optimized greedily, but this is not standard and our bound (3.10) suggests that the approximation may become worse.

### 3.2.4 Energy ($\star$) as an information criterion

Regularizers are useful energy terms because they can help to avoid over-fitting. In statistical model selection, various information criteria have been proposed to fulfil a similar role. Information criteria penalize overly-complex models, preferring to explain the data with fewer, simpler models (Occam’s razor [20]).

For example, consider the well-known Akaike information criterion (AIC) [1]:

$$\min_{\Theta} -2 \ln \Pr(X | \Theta) + 2|\Theta|$$

(3.24)

where $\Theta$ is a model, $\Pr(X | \Theta)$ is a likelihood function and $|\Theta|$ is the number of parameters in $\Theta$ that can vary. This criterion was also discussed by Torr [24] and Li [18] in the context of motion estimation.

Another well-known example is the Bayesian information criterion (BIC) [5, 20]:

$$\min_{\Theta} -2 \ln \Pr(X | \Theta) + |\Theta|\ln |P|$$

(3.25)

where $|P|$ is the number of observations. The BIC suggests that label costs should be chosen in some proportion to the number of observations (or, in our case, the expected number of observations per model). In contrast, AIC over-fits as we add more observations from the true models. See [5] for an intuitive discussion and derivation of BIC in general, and see Torr’s work [24] for insights specific to vision.

### 3.3 Working With a Continuum of Labels

Our experimental Section 3.4 focuses on multi-model fitting problems, which are the most natural applications of energy ($\star$). As was first argued in Chapter 2, energies like ($\star$) are powerful criteria for multi-model fitting in general. However, there is a technical hurdle with using combinatorial algorithms for model fitting. In such applications each label represents a specific model, including its parameter values, and the set of all labels

\(^2\)http://vision.csd.uwo.ca/code/
3.3. Working With a Continuum of Labels

![Figure 3.5](image)

(a) Ground truth. (b) Raw data. (c) Proposals (d) 1st iteration (e) 5th iteration (f) Convergence

Figure 3.5: Re-estimation helps to align models over time. Above shows 900 raw data points with 50% generated from 5 line intervals. Random sampling proposes a list of candidate lines (we show 20 out of 100). The 1st segmentation and re-estimation corresponds to Li [18], but only the yellow line and gray line were correctly aligned. The decreasing energies in Figure 3.8 correspond to better alignments like the subsequent iterations above. If a model loses enough inliers during this process, it is dropped due to label cost (dark blue line).
\( \mathcal{L} \) is a continuum. In line fitting, for example, \( \mathcal{L} = \mathbb{R}^2 \). Practically speaking, however, the combinatorial algorithms from Section 3.2 require a finite set \( \mathcal{L} \) of labels (models). Below we review a technique to effectively explore the continuum of model parameters by working with a finite subset of models at any given iteration \( t \).

**PEARL Algorithm**

| propose initial models \( \mathcal{L}_0 \) by random samples (as in RANSAC) |
| run \( \alpha \)-expansion to compute optimal labeling \( f \) w.r.t. \( \mathcal{L}_t \) |
| re-estimate model parameters to get \( \mathcal{L}_{t+1} \); \( t:=t+1; \) goto 2 |

PEARL was the first to use regularization energies and EM-style optimization for geometric multi-model fitting. Other geometric model fitting works have used separate elements such as random sampling [24, 18] (as in RANSAC) or EM-style iteration [2], but none have combined them in a single optimization framework. The experiments in Chapter 2 show that our energy-based formulation beats many state-of-the-art algorithms in this area. In other settings (segmentation, stereo) these elements have been combined in various application-specific ways [27, 2, 21, 26].

We introduce a more general energy (\( \ast \)) and a better algorithm for the expansion step of PEARL (step 2).

**Review of PEARL for (\( \ast \))**:

Step 1 of PEARL is to propose an initial set of models \( \mathcal{L}_0 \). Each proposal is generated by randomly sampling the smallest subset of data points needed to define a geometric model, exactly as in RANSAC [9]. A larger set of proposals \( \mathcal{L}_0 \) is more likely to contain models that approximate the true ones. Of course, \( \mathcal{L}_0 \) will contain many incorrect models as well, but optimizing energy (\( \ast \)) over \( \mathcal{L}_0 \) (step 2) will automatically select a small subset of labels from among the best models in \( \mathcal{L}_0 \).

The initial set of selected models can actually be further improved as follows. From here on, we represent model assignments by two sets of variables: segmentation variables \( \{f_p\} \) that for each data point \( p \) specifies the index of a model from the finite set \( \mathcal{L}_0 \), and parameter variables \( \{\theta_l\} \) that specify model parameters currently associated with each model index. Then, energy (\( \ast \)) is equivalent to

\[
E(f; \theta) = \sum_{p \in P} D_p(f_p, \theta_{f_p}) + \sum_{pq \in N} V_{pq}(f_p, f_q, \theta_{f_p}, \theta_{f_q}) + \sum_{L \subseteq \mathcal{L}} h_L(\theta_L) \cdot \delta_L(f). \tag{\( \ast \)}
\]

For simplicity, assume that the smoothness terms in (\( \ast \)) are Potts interaction potentials [4] and the third term represents simple per-label costs as in (3.1). Then, specific model parameters \( \theta_l \) assigned to a cluster of points \( P_l = \{p | f_p = l\} \) only affect the first term in (\( \ast \)), which is a sum of unary potentials. In most cases, it is easy to compute a parameter value \( \hat{\theta}_l \) that locally or even globally minimizes \( \sum_{p \in P_l} D_p(l, \theta_l) \). The re-estimated parameters \( \{\hat{\theta}_l\} \) correspond to an improved set of labels \( \mathcal{L}_1 \) that reduces energy (\( \ast \)) for fixed segmentation \( f \) (step 3).
3.4. Applications and Experimental Setup

The experimental setup is essentially the same for each application: generate proposals via random sampling, compute initial data costs $D_p$, and run the iterative algorithm.
Figure 3.8: Energy ($\star$) over time for a line-fitting example (1000 points, 40% outliers, 6 ground truth models). Only label cost regularization was used. Re-estimation reduces energy faster and from fewer samples. The first point (●) in each series is taken after exactly one segmentation/re-estimation, and thus suggests the quality of Li [18] using a greedy algorithm instead of LP relaxation.
from Section 3.3. The only components that change are the application-specific $D_p$ and regularization settings. Section 3.4.1 outlines the setup for basic geometric models: lines, circles, homographies, motion. Section 3.4.2 describes the unsupervised image segmentation setup.

### 3.4.1 Geometric multi-model fitting

Each label $l \in L$ represents an instance from a specific class of geometric model (lines, homographies), and each $D_p(l)$ is computed by some class-specific measure of geometric error. The strength of per-label costs and smooth costs were tuned for each application.

**Outliers:** All our experiments handle outliers in a standard way: we introduce a special outlier label $\phi$ with $h_{\phi} = 0$ and $D_p(\phi) = \text{const} > 0$ manually tuned. This corresponds to a uniform distribution of outliers over the domain.

**Line fitting:** Our line fitting experiments are all synthetic and mainly meant to be illustrative. Data points are sampled i.i.d. from a ground-truth set of line segments (e.g. Figure 3.5), under reasonably similar noise; outliers are sampled uniformly. Since the data is i.i.d. we set $V_{pq} = 0$ and use the greedy algorithm from Section 3.2.3. Figure 3.5 is a typical example of our line fitting result with outliers.

In 2D each line model $l$ has parameters $\theta_l = \{a, b, c, \sigma\}$ where $ax + by + c = 0$ defines the line and $\sigma^2$ is the variance of data; here $a, b, c$ have been scaled such that $a^2 + b^2 = 1$. Each proposal line is generated by selecting two random points from $P$, fitting $a, b, c$ accordingly, and selecting a random initial $\sigma$ based on a prior. The data cost for a 2D point $x_p = (x^x, x^y)$ is computed w.r.t. orthogonal distance

$$D_p(l) = -\ln \left( \frac{1}{\sqrt{2\pi}\sigma} \exp \left( \frac{-(ax^x + bx^y + c)^2}{2\sigma^2} \right) \right).$$

(3.26)

Figure 3.8 shows the trend in running time as the number of random initial proposals is increased. For 1000 data points and 700 samples, convergence took 0.7–1.2 seconds with 50% of execution time going towards computing data costs (3.26) and performing re-estimation.

**Line interval and circle fitting:** Figure 3.6 shows three interval fitting results from different initial proposals. Figure 3.7 shows a simple circle fitting result. See [8] for details.

**Homography estimation:** Our setup comes directly from Chapter 2 so we give only an outline. The input comprises two (static) images related by a fundamental matrix. We first detect SIFT features [19] and do matching as a preprocessing step; these matches are our observations. Our set of neighbours $pq \in N$ is determined by a Delaunay triangulation of feature positions in the first image. The models being estimated are homographies, and each proposal is generated by sampling four potential feature matches. Data costs measure the symmetric transfer error [11] of a match w.r.t. a homography. Figure 3.2 shows a representative result.
Figure 3.9: Unsupervised segmentation by clustering simultaneously over pixels and colour space using Gaussian mixtures (colour images) and non-parametric histograms (greyscale images). Notice we find coarser clustering on baseball than Zabih & Kolmogorov [26] without over-smoothing. For segmentation, our energy is closer to Zhu & Yuille [27] but our algorithm is more powerful than region-competition.
Multi-body motion segmentation: The setup starts the same as for homography estimation, except here each model is a fundamental matrix corresponding to a rigid body motion, as in [18], and each proposal is generated by sampling eight matches. Data costs measure the squared Sampson’s distance [11] of a match w.r.t. a fundamental matrix. Figure 3.1 shows a representative result. See Chapter 2 for details.

3.4.2 Image segmentation by MDL criterion

Here the models are either greyscale histograms or Gaussian mixtures in RGB space. Initial proposals were generated by sampling small patches of the input image, just like in [27, 26]. We used uniform Potts model for pairwise terms. See [8] for further details. Figures 3.3 and 3.9 show examples of our segmentations.

We formulate the problem as one of finding a minimum description length (MDL) representation for the image, meaning a we want to represent the image compactly, in an information-theoretic sense (see [20] for review of MDL). The MDL principle was first proposed for unsupervised segmentation by Zhu & Yuille [27], along with their region competition algorithm. When defined over a 2D grid of image pixels, our energy ($\mathbf{\star}$) can implement a discrete version of Zhu & Yuille’s energy. Our algorithm is however more powerful because $\alpha$-expansion makes large moves, while region competition relies on local contour evolution and explicit merging of adjacent regions.

3.5 Discussion

The C++ implementation and MATLAB wrapper are available online. The potential applications of our algorithm are nearly as broad as for $\alpha$-expansion. Our algorithm can be applied when observations are known a priori to be correlated, whereas standard mixture model algorithms are designed for i.i.d. data.

We can generalize the concept of label costs by making them spatially variant. The label cost term in energy ($\mathbf{\star}$) could actually be expressed as

$$\sum_{P \subseteq \mathcal{P}} \sum_{L \subseteq \mathcal{L}} h^P_L \delta_L (f_P)$$

(3.27)

where the energies discussed in this paper are the special case when $h^P_L = 0$ for all clique sets $P \subseteq \mathcal{P}$. Note that the test-and-reject approach (Section 3.2.1) to incorporate $C^a(\cdot)$ may no longer be ideal for such ‘regional’ label costs. Regional label costs may be useful when labels belong to known categories with specific priors, such as “pay a fixed penalty if any label from $\{sky, cloud, sun\}$ appears in the bottom of an image.” Indeed, our higher-order label costs themselves seem to be novel, both in vision and in terms of the UFL problem, and can be thought of as a specific type of co-occurrence cost.

Furthermore a binary construction based on Robust $P^n$ Potts [14], within our expansion step, allows us to encode an arbitrary concave penalty on the number of variables taking a specific label, thus generalizing $\delta_l (\cdot)$ if needed. We leave this as future work.

\[3\text{http://vision.csd.uwo.ca/code/}\]
Our energy is quite general but this can be a disadvantage in terms of speed. The $\alpha$-expansion step runs in polynomial time for fixed number of positive $h_L$ terms, but higher-order label costs should be used sparingly. Even the set of per-label costs $\{h_l\}$ slows down $\alpha$-expansion by 40–60%, though this is still relatively fast for such difficult energies [23]. This slowdown may be because the Boykov-Kolmogorov maxflow algorithm [3] relies on heuristics that do not work well for large cliques, i.e. subgraphs of the kind in Figure 3.4. Even if faster algorithms can be developed, our implementation can test the merit of various energies before one invests time in specialized algorithms.

Bibliography


Chapter 4

Energy based multi-model fitting & matching for 3D reconstruction

Standard geometric model fitting methods take as an input a fixed set of feature pairs greedily matched based only on their appearances. Inadvertently, many valid matches are discarded due to repetitive texture or large baseline between view points. To address this problem, matching should consider both feature appearances and geometric fitting errors. We jointly solve feature matching and multi-model fitting problems by optimizing one energy. The formulation is based on our generalization of the assignment problem and its efficient min-cost-max-flow solver. Our approach significantly increases the number of correctly matched features, improves the accuracy of fitted models, and is robust to larger baselines.

4.1 Introduction

Many existing methods for model fitting and 3D structure estimation use pre-matched image features as an input (bundle adjustment, homography fitting, rigid motion estimation). Vice versa, many matching methods (sparse/dense stereo) often use some pre-estimated structural constraints, e.g. epipolar geometry, to identify correct matches/inliers. This paper introduces a novel framework for simultaneous estimation of high-level structures (multi-model fitting) and low-level correspondences (feature matching). We discuss a regularized formulation of the proposed fit & match (FM) problem. That formulation uses a generalization of the assignment problem and we solve it using an efficient specialized min-cost-max-flow solver. This paper primarily focuses on jointly solving multi-homography fitting and sparse feature matching as a simple show case for the FM
4.1. Introduction

paradigm. Other applications would be rigid motion estimation, camera pose estimation, etc.

**Related Work:** An attempt to formulate an objective function for fitting-&-matching naturally leads to a version of the assignment problem. The majority of prior work could be divided into two major groups: matching techniques using quadratic assignment problems and FM techniques using linear assignment as sub-problems.

*Quadratic assignment problem* (QAP) normally appears in the context of non-parametric matching. For example, the methods in [16, 3, 12] estimate non-rigid motion correspondences as a sparse vector field. They rely on a quadratic term in the objective function to encourage geometric regularity between identified matched pairs. Such QAP formulations often appear in shape matching and object recognition. QAP is NP-hard and these methods use different techniques to approximate it. For example, [7] approximates QAP by iteratively minimizing its first-order Taylor expansion, which reduces to a *linear assignment problem*.

If correspondences are constrained by some parametric model(s), matching often simplifies to *linear assignment problem* when model parameters are fixed. In this case, the geometric regularity is enforced by a model fidelity term (linear w.r.t. matching variables) and pair-wise consistencies [16, 3, 12] are no longer needed. Typically for FM problems, feature matching as a *linear assignment problem* and model parameter fitting are performed in a coordinate descent fashion. For example, SoftPOSIT [5] matches 2D image features to 3D object points and estimate camera pose in such iterative fashion. Building on these ideas [15] fit a single homography using geometric and appearance priors with unknown correspondences.

Our work develops a generalization of the *linear assignment problem* for solving the FM problem when matching is constrained by an unknown number of geometric models. In contrast to [5, 15], we do not assume that matches/correspondences are constrained by a single parametric model. Note that in order to solve FM problem for multi-models, a regularization term is required to avoid over fitting. Unlike [15, 5, 16], our energy formulation includes label cost regularization as in [6].

Another related approach, *guided matching*, is a post-processing heuristic for increasing the number of matches in case of single model fitting [9]. Similar to our approach, guided matching iteratively re-estimates matches and refines the model. In contrast to our approach, guided matching pursues different objectives at refitting and re-matching steps and does not guarantee convergence. Our method could be seen as an energy-based guided matching with guaranteed convergence. Moreover, unlike guided matching [9], our regularization approach is designed for significantly harder problems where data supports multiple models.

**Contributions:** We propose a *fit-ℓε-match* energy functional (4.1)-(4.2) for jointly solving the matching and multi-model fitting problems. Our energy consists of a *unary potentials* term that describes geometric model fitting errors and feature appearance matching

---

1Geometric errors minimization vs. inliers maximization.
costs, and label cost term that discourages over-fitting by penalizing the number of models/labels assigned to matched features.

Our fit-β-match framework is based on a novel generalization of linear assignment problem to multi-model case, namely generalized assignment problem (GAP), which jointly formulates feature-to-feature matching and match-to-model assignment. We propose a fast approach for solving the regularized GAP (in which the number of models are penalized) by generalizing min-cost-max-flow techniques for bipartite weighted matching [8]. Our main technical contributions are summarized here:

- fit-β-match energy formulation (4.1)-(4.2)
- generalized assignment problem, GAP
- a fast solver for regularized GAP.

We compare our joint fitting-β-matching framework with a state-of-the-art multi-model fitting algorithm that uses pre-matched features [6] and a variant of it that uses guided matching [9]. Our approach finds more matches, estimates models more accurately, and is more robust to larger baselines between cameras.

4.2 Fit-β-Match Energy

We will use the following notations. \( \mathcal{F}_l \) and \( \mathcal{F}_r \) are the sets of observed SIFT features [13] in the left and right images, respectively. \( \mathcal{L} \) is the set of indices to all models (labels). \( \theta \) is the set of all models’ parameters, \( \theta = \{ \theta_h \mid h \in \mathcal{L} \} \) where \( \theta_h \) is the parameters of model \( h \in \mathcal{L} \). In practice, \( \theta \) could be a set of randomly sampled models, e.g. homographies. \( f \) is a labelling of all features in the left image, \( f = \{ f_p \mid p \in \mathcal{F}_l \} \) where \( f_p \) is the label assigned to feature \( p \) such that \( f_p \in \mathcal{L} \). \( x_{pq} \) is a binary variable which is 1 if \( p \) and \( q \) are matched and 0 otherwise. A matching \( \mathcal{M} \) is \( \{ x_{pq} \mid (p,q) \in \mathcal{F}_l \times \mathcal{F}_r \} \). \( Q(p,q) \) is an appearance penalty for matching features \( p \in \mathcal{F}_l \) and \( q \in \mathcal{F}_r \) based on the similarity of their descriptors.

We define fitting and matching score between features \( p \in \mathcal{F}_l \) and \( q \in \mathcal{F}_r \) for a given model \( \theta_h \) as

\[
D_{pq}(\theta_h) = ||\theta_h p - q|| + Q(p,q)
\]

combining geometric error and appearance penalty where \( || \cdot || \) denotes geometric error e.g. symmetric transfer error. A similar matching score was used in computing the ground truth matching in [14]. We use symmetric appearance penalty\(^2\) \( Q(p,q) \), e.g. the angle between the features’ descriptors of \( p \) and \( q \).

For simplification, we will introduce our energy functional under the assumption that there are no occlusions/outliers

\[
E(f, \theta, \mathcal{M}) = \sum_{p \in \mathcal{F}_l, q \in \mathcal{F}_r} D_{pq}(\theta_{f_p}) \cdot x_{pq} + \beta \sum_{h \in \mathcal{L}} \delta_h(f) 
\]  

\(^2\)In this work, \( Q(p,q) = 0 \) if the angle between two features’ descriptors is less than \( \pi/4 \) and \( \infty \) otherwise.
where
\[
\begin{align*}
\sum_{p \in F_l} x_{pq} &= 1 & \forall q \in F_r \\
\sum_{q \in F_r} x_{pq} &= 1 & \forall p \in F_l \\
x_{pq} \in \{0, 1\} & & \forall p \in F_l, \forall q \in F_r
\end{align*}
\] (4.2)
and \(\delta_h(f) = \lfloor \exists p \in F_l : f_p = h \rfloor\) and \([\cdot]\) are Iverson brackets.

**Occlusions/Outliers:** Due to occlusions we can have \(|F_l| \neq |F_r|\) which renders (4.1)-(4.2) unfeasible since the one-to-one constraints could never be met. We add |||\(|F_l| - |F_r||\) dummy features, with a fixed matching cost \(T\), to the smaller set of features to ensure feasibility. This is equivalent to changing a rectangular assignment problem to a square one. Also, to make our approach robust to outliers we introduce an outlier model \(\phi\) such that \(D_{pq}(\phi) = T\) for any \(p \in F_l\) and \(q \in F_r\). The use of an outlier model with a uniformly distributed cost \(T\) is a common technique in Computer Vision [6, 10].

**4.3 Overview of Our Approach**

Minimizing energy (4.1)-(4.2) is NP-hard. We propose to find an approximate solution by minimizing this energy in a block coordinate descent fashion.

In general, to minimize a function via block coordinate descent its coordinates (variables) are partitioned into \(n\) blocks, not necessarily mutually exclusive. At each iteration the function is sequentially minimized with respect to the coordinates in each block while fixing other coordinates that are not in this block. Intuitively, fixing different sets of variables in (4.1)-(4.2) reduces the energy to special cases which are easier to minimize.

In our case, the set of all variables \(\{f, \theta, \mathcal{M}\}\) for energy (4.1) are partitioned into two blocks \(\{f, \theta\}\) and \(\{f, \mathcal{M}\}\). Note that labelling \(f\) is in both blocks and, therefore, it is optimized at all steps.

The first block \(\{f, \theta\}\) fixes matching \(\mathcal{M}\) and energy (4.1)-(4.2) reduces to

\[
E(f, \theta) = \sum_{p \in F_l} D_p(\theta_{f_p}) + \beta \sum_{h \in \mathcal{L}} \delta_h(f)
\] (4.3)

where \(D_p(\theta_h) = D_{pq}(\theta_h)\) for all \(h \in \mathcal{L}\) provided that \(\mathcal{M}\) assigns \(q\) to \(p\), i.e. \(x_{pq} = 1\). Energy (4.3) could be efficiently minimized over \(f\) and \(\theta\) using standard multi-model fitting methods for fixed matching, e.g. PEARL [6].

The second block \(\{f, \mathcal{M}\}\) fixes parameters \(\theta\). We separately consider two cases: \(\beta = 0\) and \(\beta > 0\). In case \(\beta = 0\) the optimization problem could be optimally solved in polynomial time and the corresponding algorithm is used as a building block for the more general case \(\beta > 0\). Thus, we first discuss the simpler case \(\beta = 0\) when energy (4.1)-(4.2) does not penalize the number of models. It reduces to

\[
E(f, \mathcal{M}) = \sum_{p \in F_l, q \in F_r} D_{pq}(\theta_{f_p}) \cdot x_{pq}
\] (4.4)

subject to constraints (4.2). We will refer to (4.4)-(4.2) as the generalized assignment problem (GAP). GAP is a weighted matching problem over a fixed set of models that
match features and assigns each match to a model\(^3\). Furthermore, GAP is an integral linear program, see our proof in [11]. Section 4.4.1 describes a solver for GAP that finds its global minimum in polynomial time.

The more general case \( \beta > 0 \) of the block coordinate descent for (4.1)-(4.2) with respect to variables \( \{f,M\} \) reduces to optimization of energy

\[
E(f,M) = \sum_{p \in F_i, q \in F_r} D_{pq}(\theta_{fp}) \cdot x_{pq} + \beta \sum_{h \in L} \delta_h(f)
\]

subject to constraints (4.2), which is NP-hard. We will refer to (4.5)-(4.2) as Regularized-GAP (Reg-GAP). Section 4.4.2 introduces Local Search-GAP (LS-GAP) approximation algorithm for energy (4.5)-(4.2). It uses our GAP solver in a combinatorial local search fashion iteratively exploring different subsets of models and selecting solutions reducing energy (4.5) which requires solving a series of similar GAP instances efficiently.

Our Energy-based Fitting & Matching (EFM) algorithm for energy (4.1)-(4.2) can be summarized as

**Energy-based Fitting & Matching (EFM)**

**Initialization:** Find an initial matching \( M \) using standard matching techniques

repeat

1-Given matching \( M \), minimize (4.3) using PEaRL [6] to find labelling \( f \) and models’ parameters \( \theta \).
2-Given models’ parameters \( \theta \), minimize (4.5)-(4.2) using LS-GAP, see Sec. 4.4.2, to find matching \( M \) and labelling \( f \).

until energy (4.1) converges

EFM finds an initial matching using standard matching techniques, e.g. standard SIFT matching [13]. Then, it iteratively optimizes energy (4.1)-(4.2) by alternatively minimizing energy (4.3) over labelling \( f \) and models’ parameters \( \theta \) for fixed matching \( M \), and minimizing energy (4.5)-(4.2) over \( f \) and \( M \) for fixed \( \theta \). Although EFM is guaranteed to converge since energy (4.1) is bounded below, i.e. (4.1) \( \geq \beta \), it is not trivial to derive a theoretical bound on the convergence rate and approximation ratio for EFM. However, in Section 4.5, we empirically show that EFM converges in a few iterations to a near optimal solution.

### 4.4 Optimization

Section 4.4.1 shows how to optimally solve GAP (4.4)-(4.2) and a series of similar GAP instances, efficiently. In Section 4.4.2 we introduce the Local Search-GAP (LS-GAP) algorithm for finding an approximate solution for energy (4.5)-(4.2)—that requires solving many similar GAP instances sequentially.

\(^3\)Our definition of GAP is different from some generalizations of the assignment problem in the optimization literature.
4.4. Optimization

4.4.1 Solving GAP

There are alternative ways to solve GAP [11]. This section describes an approach that we find most efficient. We reduce GAP to the standard Linear Assignment Problem (LAP) and propose an efficient solver for sequences of similar problems.

Reducing GAP to LAP

GAP (4.4)-(4.2) reduces to LAP since \( f \) and \( M \) are independent: any pair \((p,q)\) has optimal label

\[
    f_p = \arg\min_{h \in \mathcal{L}} D_{pq}(\theta_h)
\]

independently from the value of \( x_{pq} \). A simple proof by contradiction shows that the previous statement is true. Given an optimal GAP solution where \( x_{pq} \) was assigned to label \( k \) such that \( D_{pq}(\theta_k) > \min_{h \in \mathcal{L}} D_{pq}(\theta_h) \) then the solution is not optimal as we could decrease the energy by assigning \( x_{pq} \) to model \( k^* \) without violating any of the linear constraints where

\[
    k^* = \arg\min_{h \in \mathcal{L}} D_{pq}(\theta_h).
\]

In other words, unregularized GAP could be reduced to a regular assignment problem by selecting the model with lowest cost for every possible match. After computing the optimal labeling \( f \), the optimal matching \( M \) in (4.4)-(4.2) could be found by solving the following LAP

\[
    E(M) = \sum_{p \in \mathcal{F}_l, q \in \mathcal{F}_r} D_{pq} \cdot x_{pq}
\]

subject to (4.2) where

\[
    D_{pq} = \min_{h \in \mathcal{L}} D_{pq}(\theta_h).
\]

LAP as MCMF (overview)

LAP (4.6)-(4.2) can be equivalently formulated as a standard min-cost-max-flow (MCMF) problem with known efficient solvers [1]. This problem is defined as follows. Let \( G = (V, E) \) denote a graph with vertices \( V \) and edges \( E \) where each edge \((v,w) \in E\) has a capacity \( u(v,w) \) and cost \( c(v,w) \). Let \( F \) be a flow function such that \( 0 \leq F(v,w) \leq u(v,w) \) for over all edges in \( E \). The cost of an arbitrary flow function \( F \) is defined as \( \text{cost}(F) = \sum_{(v,w) \in E} c(v,w) \cdot F(v,w) \). MCMF is a valid maximum flow \( F \) from \( s \) to \( t \) in \( V \) that has minimum cost.

To formulate LAP (4.6)-(4.2) as MCMF problem we build graph \( G=(V, E) \) with nodes

\[
    V = \{s, t\} \cup \{p \mid p \in \mathcal{F}_l\} \cup \{q \mid q \in \mathcal{F}_r\},
\]

edges

\[
    E = \{(s,p), (q,t), (p,q) \mid p \in \mathcal{F}_l, q \in \mathcal{F}_r\},
\]

capacity \( u(v,w) = 1 \) for all edges \((v,w) \in E\), and cost \( c(p,q) = D_{pq} \) for edges \((p,q) \in \mathcal{F}_l \times \mathcal{F}_r \) and 0 for other edges. The optimal \( M \) and \( f \) for GAP can be obtained from MCMF flow \( F^* \) for \( G \) as \( x_{pq} = F^*(p,q) \) for all \((p,q) \in \mathcal{F}_l \times \mathcal{F}_r \) and

\[
    f_p = \arg\min_{h \in \mathcal{L}} D_{pq}(\theta_h)
\]

if \( p, q \) are matched, \( x_{pq} = 1 \).
Chapter 4. Energy based multi-model fitting & matching

Solving MCMF (overview)

There are many algorithms for finding MCMF for a given graph [1]. We overview the Successive Shortest Path (SSP) algorithm [1] in order to introduce our flow recycling technique for efficiently solving similar GAP instances. SSP successively finds the shortest path w.r.t. edge costs from $s$ to $t$ and augments these paths until the network is saturated. For unit capacity graphs, augmentation of an edge reverses its direction and flips its cost sign. Finding the shortest path with negative costs is expensive. Instead of the original costs SSP uses reduced costs $c^\pi(v, w) := c(v, w) - \pi(v) + \pi(w) \geq 0$ where $\pi(v)$ is the potential of node $v$. Initially set to zero, node potentials are updated after each path augmentation to ensure that the reduced costs non-negativity constraints are satisfied, see [1] for more details. Let $n = |\mathcal{F}_r| = |\mathcal{F}_l|$ be the number of features. A shortest path w.r.t. $c^\pi$ could be found in $O(n^2)$ using Dijkstra’s algorithm. By construction, we need to find $n$ paths. Thus, SSP is $O(n^3)$ when solving LAP.

Solving a Series of GAPs

We propose an $O(n^2)$ method for solving MCMF corresponding to a modified LAP (4.6)-(4.2) after changing one or all edge costs associated with one feature in $\mathcal{F}_l$. Assume MCMF $\mathcal{F}$ for $\mathcal{G}$ and node potential function $\pi$ that satisfy the reduced costs non-negativity constraints on the residual graph $\mathcal{G}_r$. Changing edge costs associated with feature $p$ may violate reduced cost non-negativity constraints involving $p$. To regain feasibility after dropping the no longer needed artificial nodes $s$ and $t$ and their edges, we reverse the flow through $(p, q)$ where $p$ and $q$ are matched by $\mathcal{F}$ and update $\pi(p)$

$$\pi(p) = \min c(p, v) + \pi(v) \quad \forall v \in \mathcal{F}_r.$$ 

Finally, we push one unit of flow from $p$ to $q$, i.e. find the shortest path w.r.t. $c^\pi$, to maximize the flow. The reduced cost optimally theorem [1] grantees that the resulting flow is MCMF. In case $m$ features in $\mathcal{F}_l$ had their associated costs changed, the new MCMF could be found in $O(mn^2)$ by applying the steps above sequentially to each feature. These steps could be used with any LAP [4] or MCMF solver not just SSP. Given an optimal solution for LAP (4.6)-(4.2), it is possible to compute optimal node potentials that satisfy reduced cost non-negativity constraints in polynomial time [1, 11].

4.4.2 Local Search-GAP (LS-GAP)

Now we introduce a local search algorithm that solves a regularized GAP (4.5)-(4.2) using the GAP algorithm introduced in Section 4.4.1 as a sub-procedure. Assume that $\mathcal{L}$ is the current set of possible models$^4$. Let $\mathcal{L}_c$ be an arbitrary subset of $\mathcal{L}$ and $\mathcal{M}_f(\mathcal{L}_c)$ denote the GAP solution when the label space is restricted to $\mathcal{L}_c$. Note that GAP ignores the label cost term in (4.5) but we could easily evaluate energy (4.5) for $\mathcal{M}_f(\mathcal{L}_c)$. The proposed LS-GAP algorithm heuristically searches over different subsets $\mathcal{L}_c \in \mathcal{L}$ for one such that $\mathcal{M}_f(\mathcal{L}_c)$ has the lowest value of energy (4.5). Our motivation to search for

---

$^4$In practice, we restrict $\mathcal{L}$ to be the set of models that were assigned to at least one matched pair of features in (4.3) solution.
minima of (4.5)-(4.2) only among GAP solutions comes from an obvious observation that a global minima of (4.5)-(4.2) must also solve the GAP if the label space is restricted to the right subset of $\mathcal{L}$.

We define sets of all possible add, delete and swap combinatorial search moves as

$$
\mathcal{N}^a(\mathcal{L}_c) = \bigcup_{h \in \mathcal{L} \setminus \mathcal{L}_c} \{ \mathcal{L}_c \cup h \}
$$

$$
\mathcal{N}^d(\mathcal{L}_c) = \bigcup_{h \in \mathcal{L}_c} \{ \mathcal{L}_c \setminus h \}
$$

$$
\mathcal{N}^s(\mathcal{L}_c) = \bigcup_{h \in \mathcal{L}_c} \{ \mathcal{L}_c \cup \ell \setminus h \}
$$

These are three different local neighbourhoods around $\mathcal{L}_c$. We also define a larger neighbourhood $\mathcal{N}^\ast$ around $\mathcal{L}_c$ as

$$
\mathcal{N}^\ast(\mathcal{L}_c) = \mathcal{N}^a(\mathcal{L}_c) \cup \mathcal{N}^d(\mathcal{L}_c) \cup \mathcal{N}^s(\mathcal{L}_c)
$$

LS-GAP uses a combination of add, delete and swap moves, similar to the work in [2], to heuristically find a set of labels near current set $\mathcal{L}_t$ that is better w.r.t. energy (4.5).

<table>
<thead>
<tr>
<th>Local Search-GAP (LS-GAP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{L}_t \leftarrow \phi$, $\mathcal{N}_t \leftarrow \mathcal{N}^\ast(\mathcal{L}_t)$</td>
</tr>
<tr>
<td>while $\exists \mathcal{L}_c \in \mathcal{N}_t$</td>
</tr>
<tr>
<td>if energy (4.5) of $\mathcal{M}_f(\mathcal{L}_c)$ &lt; energy (4.5) of $\mathcal{M}_f(\mathcal{L}_t)$</td>
</tr>
<tr>
<td>$\mathcal{L}_t \leftarrow \mathcal{L}_c$, $\mathcal{N}_t \leftarrow \mathcal{N}^\ast(\mathcal{L}_t)$</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>$\mathcal{N}_t \leftarrow \mathcal{N}_t \setminus \mathcal{L}_c$</td>
</tr>
<tr>
<td>return GAP solution $\mathcal{M}_f(\mathcal{L}_t)$</td>
</tr>
</tbody>
</table>

## 4.5 Evaluation

In this section, we discuss some of the EFM framework properties, e.g. convergence rate etc. Then we compare the matching quality of our proposed EFM framework to standard SIFT matching [13]. We also compare the matching quality and the accuracy of models estimated by the EFM framework, Energy-based multi-model Fitting (EF) algorithm [6, 10], and EF followed by guided matching [9] (EF+GM). Carrying out these experiments requires knowing the ground truth of the dataset at hand. We computed ground truth; matching $\mathcal{M}_{GT}$, model estimates $\theta_{GT}$ and labelling $f_{GT}$ as described in [11].

The effect of EFM iterations on energy (4.1) for different $|\mathcal{L}|$ is shown in Fig. 4.1(a). For each $|\mathcal{L}|$ the experiment is repeated 50 times. On average each iteration took 1 min., and most of the energy was reduced in the first three iterations. EFM converged on average after 5 iterations.

EFM is non-deterministic as it uses a set of randomly sampled models $\mathcal{L}$. Figure 4.1(b) shows final energy histograms to different sizes of $\mathcal{L}$. As shown the bigger $|\mathcal{L}|$ is the more likely the final energy is going to be small, i.e. better solutions. Using a large $|\mathcal{L}|$ helps EFM avoid local minima.
Our matching evaluation criterion is based on Receiver Operating Characteristics (ROC) of the True Positives Rate vs. the False Positives Rate. The ROC attributes for an estimated matching \( M \) and ground truth matching \( M_{GT} \) are defined as follows: *Positives (P)* number of matches in \( M_{GT} \), *Negatives (N)* number of potential matches that were rejected by \( M_{GT} \), *True Positives (TP)* number of matches in \( M \) and \( M_{GT} \) (intersection), *False Positives (FP)* number of matches in \( M \) but not in \( M_{GT} \), True Positives Rate (TPR) \( \frac{TP}{P} \), and False Positives Rate (FPR) \( \frac{FP}{N} \).

A basic comparison between the matching quality of EFM and standard SIFT matching is shown in Fig. 4.2. The ROC curve, in Fig. 4.2, of SIFT matching [13] is achieved by varying the threshold on the second best ratio (SBR) \(^5\). For EFM we show a scatter plot since it is non-deterministic. We also related the EFM scatter plot to the achieved final energy by color coding it. As can be seen, for EFM the lower the final energy (blue is low energy) the better the matching quality. Also, EFM outperformed SIFT matching by reaching high TPR values.

The plots in Fig. 4.1 and 4.2 are shown for Oxford’s Merton College example of Fig. 4.3(b), to illustrate the characteristics/behavior of the EFM algorithm. Note that it will be meaningless to average these plots over different examples as they would not share the same energy scale.

For measuring the accuracy of an estimated model \( \theta_h \), we used the following geometric error ratio

\[
GQ(\theta_h) = \frac{STE(\theta_h, f_{GT}, M_{GT})}{STE(\theta_{GT}, f_{GT}, M_{GT})}
\]

where \( STE(\theta_h, f, M) \) is the Symmetric Transfer Error of \( \theta_h \) computed for labeling \( f \) and matching \( M \). A perfect model estimate will have \( GQ = 1 \).

Table 4.1 shows the effect of increasing the cameras’ baseline on the quality of estimated models and matching for EFM, EF and EF+GM methods. For small baseline, EFM and EF+GM results were comparable. For larger baselines, unlike the EF and EF+GM, the EFM model estimates’ accuracy and matching quality did not deteriorate. In general, EF+GM was prone to higher false positive rates compared to EFM.

As a multi-model show case example, Fig. 4.3(a) and (b) show the labeling \( f \) result of EF and EFM on one of the stereo images, respectively. EFM, on the average of 50 runs, found double the number of matches compared to EF which takes SIFT matching as an input. Figures 4.3(c-f) are the enlargements of the segments shown in (a) and (b) as white rectangles. Figures 4.3(g) and (h) show part of the feature matching between the left and right images of the EF and EFM results, respectively.

More results are shown in Fig. 4.4. In general, EFM found more matches, but in particular, EFM outperformed EF in two examples: graphite example (second row) with a the large baseline between camera positions, and redbrick house example (third row) with repetitive texture reduced the discriminative power of SIFT. EFM found approximately 5% to 8% more matches than EF+GM. The EF+GM results are not shown as they where visually similar to EFM.

---

\(^5\)SBR is the ratio of the distance between a left feature descriptor and its closest right feature descriptor to the distance of its second closest
Figure 4.1: Best seen in Color, Fig. (a) shows the effect of EFM iterations on energy (4.1). EFM converged on the average after 5 iterations, and each iteration on the average took 1 min. Figure (b) shows multiple histograms of the final energies for different sizes of initial set of proposals—blue indicates a large set of proposals. The larger the set of initial proposals $\mathcal{L}$ the more likely that EFM will converge to a low energy.
Figure 4.2: Best seen in Color, Fig. shows ROC curve for standard SIFT matches by varying SBR threshold, and the scatter plot represents EFM results for different sizes of initial set or proposals. As can be seen, the lower the final energy (blue) the better the matching.

<table>
<thead>
<tr>
<th></th>
<th>GQ</th>
<th>ROC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>med.</td>
<td>mean</td>
</tr>
<tr>
<td><strong>small</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFM</td>
<td>1.01</td>
<td>1.01</td>
</tr>
<tr>
<td>EF</td>
<td>1.04</td>
<td>1.05</td>
</tr>
<tr>
<td>EF + GM</td>
<td>1.01</td>
<td>1.02</td>
</tr>
<tr>
<td><strong>medium</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFM</td>
<td>1.02</td>
<td>1.02</td>
</tr>
<tr>
<td>EF</td>
<td>1.20</td>
<td>1.30</td>
</tr>
<tr>
<td>EF + GM</td>
<td>1.07</td>
<td>1.12</td>
</tr>
<tr>
<td><strong>large</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFM</td>
<td>1.05</td>
<td>1.07</td>
</tr>
<tr>
<td>EF</td>
<td>1.90</td>
<td>2.24</td>
</tr>
<tr>
<td>EF + GM</td>
<td>1.49</td>
<td>1.78</td>
</tr>
</tbody>
</table>

Table 4.1: Graphite VGG Oxford dataset, single model and increasing baseline. The table shows the averages of GQ and ROC attributes, over 50 runs, for EFM, EF, and EF+GM model estimates.
Figure 4.3: Best seen in Color, Fig. (a) shows EF labeling result (average TPR=0.51 and FPR=1.6E-05) and (b) shows EFM labeling result (average TPR=0.98 and FPR=9.1E-06). Features assigned to the same model/label are drawn in the same color and unmatched features are shown as white x. Figures (c-f) show the enlargement of segments 1 and 2 in (a) and (b). Figures (g-h) show the matching, between two small regions in the stereo images, of the EF and EFM results, respectively.
Figure 4.4: Best in Color, the first column shows one of the stereo images for each example, second and third columns show the EF and EFM labelling results superimposed on the images shown in the first column, respectively. On average of 50 runs, EFM found 0.75, 10.53, 3.31, 0.44, and 0.68 times more inliers than EF. EFM and EF+GM results where comparable, EFM found approximately between 0.05 and 0.08 times more matches.
4.6 Conclusions

We introduced regularized energy functional that jointly formulates multi-model fitting and matching problems, and a framework to optimize it. Our results show that the framework finds near optimal matching, and when compared to state-of-the-art multi-model fitting algorithm our framework finds better models’ estimates and it is more robust to large baselines. We also showed how to efficiently find optimal feature-to-feature matching and match-to-model assignment for a given set of models with label cost. Furthermore, our framework can be used with more complex models, e.g. fundamental matrices, without affecting the framework’s complexity, unlike [15]. Currently, our framework requires an initial matching for future work we aim to alleviate the need for it.

Bibliography


Chapter 5

Conclusion

We proposed a new general approach to geometric multi-model fitting where we formulated the problem as a discrete labeling of data points using MRF and MDL style regularization functionals widely used in other computer vision problems. The goal is to find models “explaining” all data points based on spatial regularity and sparsity priors. The continuous space of model parameters is explored via PEARL algorithm that combines data sampling and energy minimization which iterates over labeling and re-estimation optimization steps in a block coordinate descent fashion. The method automatically obtains a small number of models that “explain” data. Many empirical tests on synthetic and real imagery demonstrate a strong potential of our general approach and its applicability to a wide spectrum of multi-model fitting problems.

The $\alpha$-expansion algorithm has had a significant impact in computer vision due to its generality, effectiveness, and speed. It is commonly used to minimize energies that involve unary, pairwise, and specialized higher-order terms. PEARL uses $\alpha$-expansion when it utilizes only the spatial regularizer while the newly proposed $\alpha$-expansion extension is used when the spatial and sparsity regularizers are considered. The proposed $\alpha$-expansion
extension optimizes label costs with well characterized optimality bounds. Label costs penalizes the number of distinct labels appearing in a solution. Our experimental results show that our $\alpha$-expansion extension converges to better solutions compared to state-of-the-art regularized multi-model fitting frameworks.

Furthermore, we introduced regularized energy functional that jointly formulates multi-model fitting and matching problems, and a framework to optimize it. Our results show that the framework finds near optimal matching, and when compared to state-of-the-art multi-model fitting algorithm our framework finds better models’ estimates and more robust to large baselines and scenes with repetitive texture. We also showed how to efficiently find feature-to-feature matching and match-to-model assignment for a given set of models with label cost. Also, our framework can be used with more complex models, e.g. fundamental matrices, without affecting the overall framework’s complexity. Currently, our framework requires initial matching for future work we aim to alleviate the need for it. For example, model proposals could be found by discretizing the space of the model parameters’ space or randomly generating model proposals using the set of all possible matches.
Curriculum Vitae

Name: Hossam Isack

Post-Secondary Education and Degrees:
Alexandria University, Egypt
2001 - 2006 BSc (Hons)

University of Western Ontario
2008 - 2009 MSc

Honours and Awards:
Ontario Graduate Scholarship
2013-2014

Related Work Experience:
Research Consultant, Imagien8, Canada
2013 - 2014

Teaching and Research Assistant
University of Western Ontario
2008 - 2014

Publications:

Journals


Conferences
