Topics in Motion Analysis

Tal Nir
Topics in Motion Analysis

Research Thesis Submitted in Partial Fulfillment
of The Requirements for the Degree of Doctor of Philosophy.

Tal Nir

Submitted to the Senate of
the Technion - Israel Institute of Technology

TEVET, 5767
HAIFA
DECEMBER 2006
This Research Thesis Was Done Under the Supervision of

Prof. Alfred Bruckstein

in the Department of Computer Science

THE GENEROUS FINANCIAL HELP OF THE TECHNION IS
GRATEFULLY ACKNOWLEDGED.

I would like to thank my instructor Prof. Freddy Bruckstein who helped and encouraged me throughout my studies even at hard times when success seemed far beyond the horizon and who inspired me in his rare combination of a scientist and an artist.

I would like to thank Prof. Ron Kimmel who although was not an official instructor had an enormous contribution to this work. Our conversations motivated me to succeed.

To my wife Rotem who took care to shake my life so much that I did not forget I’m alive.

Finally I would like to thank my grandmother Aya, without her help and encouragement I would not have even started my first degree, not mentioning the third.
## Contents

### Chapter

1. **Introduction**
   1.1 Motivation .............................. 3  
   1.2 Main results and contribution ........ 4  
   1.3 Related work ........................... 5  

2. **Joint Optical Flow Computation and sequence Denoising**
   2.1 Introduction ............................. 9  
   2.2 Problem Formulation ..................... 10  
      2.2.1 Traditional optical flow functionals ................................................................ 11  
      2.2.2 Joint optic-flow and video restoration ......................................................... 11  
      2.2.3 Specific choice for the functional ................................................................. 13  
   2.3 Implementation details ................. 16  
   2.4 Experimental Results .................... 18  
      2.4.1 Yosemite sequence .................... 18  
      2.4.2 Office sequence ....................... 22  
   2.5 Summary .................................... 24  

3. **Model Based Total Variation Regularization**
   3.1 Introduction ............................... 25  
   3.2 Problem formulation ...................... 26  
   3.3 Total variation of piecewise parametric functions ............................................ 31  
   3.4 An over-parameterized model for the 1D total variation functional .................. 32  
      3.4.1 Implementation details for the 1D examples ............................................... 35  
   3.5 Over-Parameterized image representations for total variation image denoising ........ 35  
   3.6 Summary .................................... 42
### Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Yosemite without clouds. Optical flow alone.</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>Yosemite without clouds. Coupled solution of optical flow and sequence denoising. Left: $\lambda = 0.4/\text{NoiseStandardDeviation}$. Right: constant $\lambda = 0.01$.</td>
<td>19</td>
</tr>
<tr>
<td>2.3</td>
<td>Yosemite without clouds. Results reported in [10].</td>
<td>20</td>
</tr>
<tr>
<td>2.4</td>
<td>Yosemite without clouds. Results reported in [12].</td>
<td>20</td>
</tr>
<tr>
<td>2.5</td>
<td>Our results on the office sequence. Left - coupled solution. Right - optic flow alone.</td>
<td>22</td>
</tr>
<tr>
<td>2.6</td>
<td>Office sequence. AAE Results from the literature LK=Lucas and Kanade. HS=Horn and Schunck. CLG= Combined local global approach [12].</td>
<td>23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Parameter settings.</td>
<td>56</td>
</tr>
<tr>
<td>4.2</td>
<td>Yosemite sequence without clouds.</td>
<td>58</td>
</tr>
<tr>
<td>4.3</td>
<td>Yosemite without clouds - noise sensitivity results.</td>
<td>58</td>
</tr>
<tr>
<td>4.4</td>
<td>Synthetic piecewise affine flow test.</td>
<td>64</td>
</tr>
<tr>
<td>5.1</td>
<td>Scene fiducial geometry.</td>
<td>85</td>
</tr>
</tbody>
</table>
Figures

Figure

2.1 Yosemite sequence - noise sensitivity results. ................. 20
2.2 Frame 8. Left-original. Middle-image with noise STD of 40. Right-denoised image. ........................................ 21
2.3 Optical flow of the Yosemite sequence (excluding the sky region). Left - Ground truth. Right - results obtained by the coupled solution with noise standard deviation of 40. .................. 21
2.4 Frame 10 of the office sequence. From left to right: Original, with noise STD=40, and Denoised. .............................. 23
3.1 Example 1 with $L_2$ data term and $L_2$ regularization. .... 28
3.2 Example 1 with $L_2$ data term and $L_1$ regularization. .... 28
3.3 Error graph of example 1 with $L_2$ data term and $L_1$ regularization, showing the influence of a single outlier. ....................... 29
3.4 Error graph of example 1 with $L_1$ data term and $L_1$ regularization, showing the negligible influence of a single outlier in this case. 30
3.5 Piecewise quadratic function of example 2. ......................... 36
3.6 Example 2: reconstruction errors for the regular (dotted) versus linear over-parameterized total variation for the noise free case. 37
3.7 Example 2: reconstruction errors for the regular (dotted) versus linear over-parameterized total variation for noise STD=0.01. 38
3.8 Example 2: reconstruction errors for the regular (dotted) versus linear over-parameterized total variation for noise STD=0.05. 39
3.9 Prisoner’s room. Left - Ideal image. Right - Dark photo with noise STD=5.1. ...................................................... 42
3.10 Prisoner’s room. Left - Stretched image. Middle - Image after total variation denoising. Right - Image after total variation denoising with the over-parameterized model. .................... 43
3.11 Error image (the same linear scaling from errors to gray-levels was used for both images. Left - Errors for the regular total variation. Right - Errors for the over-parameterized total variation. ....... 43
4.1 Optic flow of the Yosemite sequence. Left - ground truth. Right - Our method with the pure translation model. .......................... 59
4.2 Images of the angular error (sky region excluded), white indicates zero error and black indicates an error of 3 degrees or above. Left - our method with the affine model. Middle - our method with the pure translation model. Right - the method of [10], optic flow field courtesy of the authors. .................................................. 59
4.3 Histogram (left) and Cumulative probability (right) of the angular errors. Solid - pure translation model. Dash dot - Affine model. Dotted - Optic flow obtained in [10] courtesy of the authors. ...... 60
4.4 Solution of the affine parameters for the Yosemite sequence - from left to right $A_1, A_2, A_3$. ......................................................... 60
4.5 Solution of the affine parameters for the Yosemite sequence - from left to right $A_4, A_5, A_6$. ......................................................... 61
4.6 Solution of the pure translation parameters for the Yosemite sequence - from left to right $A_1, A_2, A_3$. ......................................................... 61
4.7 The two images: Left - first frame. Right - second frame. ....... 63
4.8 Synthetic piecewise affine flow - Ground truth. Left - $u$ component. Right - $v$ component. ......................................................... 64
4.9 Solution of the affine parameters - from left to right $A_1, A_2, A_3$. 64
4.10 Solution of the affine parameters - from left to right $A_4, A_5, A_6$. 64
4.11 Ground truth for the synthetic test (left) and computed flow by the affine over-parametrization model (right). ................................. 65
4.12 Flower garden sequence: First image (left). Optical flow computed by the 2D affine over-parametrization model, $u$ (middle) and $v$ (right). ......................................................... 66
4.13 Flower garden sequence: Optical flow computed by the 2D pure translation over-parametrization model, $u$ (left) and $v$ (right) ....... 67
4.14 Flower garden sequence: The solution of the pure translation parameter. From left to right $A_1, A_2, A_3$. ........................................ 67
4.15 Road sequence. Frame 1 - Left. Frame 2 - Right. .................. 68
4.16 Optical flow of the road sequence with the affine over-parameterization model. $u$ is shown on the left and $v$ on the right. ............... 69
4.17 Difference of the frames. Left - Difference of the original frames ($PSNR = 23.73dB$). Right - Motion compensated difference by the estimated optical flow ($PSNR = 38.17dB$). Both images are scaled by the same linear transformation from errors to graylevels. 69

5.1 Scheme for partially decoupled SLAM. ................................. 80
5.2 The translation error with no outliers in the data. .................... 82
5.3 The rotation error with no outliers in the data. ........................ 83
5.4 The rotation errors with 33 percent of outliers. ........................ 84
5.5 First frame. ......................................................... 86
5.6 Estimated camera 3D translation using the proposed approach. . 90
5.7 Estimated camera 3D rotation using the proposed approach. . . 91
5.8 Estimated camera 3D translation using the EKF approach. . . . 92
5.9 Estimated camera 3D rotation using the EKF approach. . . . . . 93
5.10 Estimated camera 3D translation using only the inlier points with
the proposed approach. ............................................ 94
5.11 Estimated camera 3D rotation using only the inlier points with the
proposed approach. .............................................. 95
5.12 Estimated camera 3D translation using only the inlier points with
the EKF approach. .............................................. 96
5.13 Estimated camera 3D rotation using only the inlier points with the
EKF approach. ................................................. 97
5.14 Image plane errors. Robust approach showing the 6 inliers. . . 98
5.15 Image plane errors. Robust approach showing the 3 outliers. . . 99
5.16 Image plane errors. EKF approach with all 9 features. ........... 100
Abstract

In this work we cover several motion analysis topics. In the context of optical flow estimation and sequence restoration, we show that the two problems are coupled and should be solved simultaneously, we formulate a single functional which expresses this coupling. The resulting numerical scheme iteratively changes both the optical flow and the image sequence and achieves improved optical flow reconstruction under noise. In the context of the total variation denoising, we propose an over-parameterized model based approach which yields improved regularization by penalizing for deviations from the desired model. We derive the Euler-Lagrange equations in the higher dimensional over-parameterized space. The proposed overparameterization methodology developed here is a general tool which can be used in many problems requiring the use of regularization. Then, in the context of optical flow, we express the optical flow by a general over-parameterized model which can represent models with independent coefficients for the $u$ and $v$ components (such as the affine case) or models with coupling between the coefficients as in the case of a rigid or pure translation motion models. Finally, we address the problem of camera motion estimation employing the CONDENSATION methodology which was originally developed for contour tracking and tools from robust statistics yielding an efficient and robust scheme for camera motion estimation.
Chapter 1

Introduction

1.1 Motivation

Video image sequences enable us to see the world in motion. The images vary in time due to camera movements and/or to the presence of moving objects in the scene. Even still images can capture motion through the effect of motion blur. This effect is sometimes used at the extreme by photographers in long exposure photos of city streets at night, long red and white traces capturing the light of cars in motion creating spectacular effects. Motion has high importance for humans and animals hence the brain developed advanced motion detection and analysis capabilities. Motion is useful for understanding the geometric structure of a scene and for finding small objects which move differently from the background. In the technological fields of image analysis, motion is important for several tasks. In video compression, a predictor for the next image is constructed using previous frames and the motion information. In robotics applications, motion enables segmenting objects in the scene by their different motion characteristics. Super-resolution procedures construct high resolution and high quality images from several low resolution images and this process requires high accuracy motion estimation in order to properly register the images at sub-pixel
accuracy.

1.2 Main results and contribution

The objective of this research was to find ways of improving motion estimation techniques thorough a revision of the underlying variational methodology used for solving these problems. As a result of revisiting and analyzing the previous literature on this topic:

(1) We suggest a coupled solution of the optical flow estimation problem and image sequence restoration. The motivation for this stems from the data term of the optical flow functional. We rely on modern variational optical flow techniques with their numerical schemes and add an additional fidelity term to form a unified functional which is minimized by solving jointly for the optical flow field and the reconstructed image sequence.

(2) We propose a new regularization strategy in conjunction with general variational approaches based on increasing the dimensionality of the problem using a model based over-parametrization. As a consequence we derive a new total variation approach which is a generalization of the classical formulation, based on the over-parametrization idea. The Euler-Lagrange equations are derived for the general model and form the basis for the numerical optimization. We show its application on 1D functions and for images (functions in 2D).

(3) We propose the application of the over-parameterized model to the representation and regularization of optical flow. The Euler-Lagrange equations for the general model and several useful examples are analyzed. Then
state of the art numerical schemes are adapted for the over-parameterized model yielding the currently best optic flow estimation results on the famous Yosemite (without clouds) benchmark sequence.

(4) We adapt the CONDENSATION framework originally used for contour tracking for the problem of camera motion estimation. The estimation state vector is composed of 12 components consisting of the camera translation and rotation and the corresponding velocities. The estimation is performed by propagating the probability density function which is approximated by samples of the state vector and their corresponding probabilities via a motion model and a measurement process. The measurement process uses image features in order to evaluate the relative probability of each hypothesis (state vector). In the measurement process, we adopt a robust statistics based measure in order to significantly reduce the influence of outliers in the feature tracking data.

1.3 Related work

In this section we briefly survey some of the basic background literature, and a more detailed discussion is provided in each section of the thesis.

1.3.0.1 Optical Flow Estimation

Optical flow is extensively studied in the field of computer vision for many years. Synthetic test image sequences were generated with ground truth for optical flow performance testing, such as the Office scene and the famous Yosemite benchmark sequence which simulates a fly through a 3D model of the Yosemite nature reserve (see [5]). Measures for quantifying the errors between the computed
flow and the ground truth were also established in order to enable comparison and ranking of the performance of different methods. A comparative study using these measures is presented in [4]. The benchmark sequences boosted competition and motivated continuous improvements of the results. Indeed, tremendous improvements of the results were accomplished over the years due to a thorough understanding of causes for errors in the assumptions and numerical schemes of the various methods proposed. Two important developments were: (1) Using multi-scale which helps to avoid falling into local minima (2) Solving for the exact brightness constancy equation instead of the linearized one helps to improve the accuracy of the solution especially when the flow field is large (see for example [10]). The pioneering works that established two different basic approaches for optical flow computation were: (1) The work of Lucas and Kanade [34] who offered an efficient numerical scheme to solve motion by locally assuming a constant motion model and (2) The work by Horn and Schunck [25] who formulated the optical flow estimation problem in a variational framework. In recent years, the variational methods were the subject of most research efforts in this field due to their superior results and also because they enable computation of a full optical flow field of the scene.

1.3.0.2 CONDENSATION

The CONDENSATION framework is based on the fundamentals of non-linear estimation methods. It was established by Isard and Blake [29] for the problem of contour tracking. The method represents the evolution of probability density functions for the state of a dynamic process via statistical samples of the state vectors, where the higher probability states are assigned more samples (known also
as importance sampling). The states in the work of [29] consisted of the image coordinates of the points forming a 2D spline representation of the contour. The samples are then propagated by a dynamic model and then a measuring phase is performed by quantifying the distance between the hypothesis contour (described by the spline for a specific state sample) and the closest edge point in the image along the normal to the contour. The measurement assigns a probability to each state (particle) which is then used for the probability of re-sampling this state in the next phase. In order to achieve robustness to outliers (missing contour edges in the image), the distance at each contour point is truncated from above at an appropriate threshold to form a truncated quadratic. The CONDENSATION framework reduces to the Kalman filter in the case of a linear dynamical model with Gaussian distribution of the disturbance and measurement noise. The CONDENSATION framework can easily handle larger variety of estimation problems with non Gaussian noise, non-linear dynamics and non-linear measurement processes.
Chapter 2

Joint Optical Flow Computation and sequence Denoising

2.1 Introduction

Optical flow computation is probably as old as computer vision. It is useful for various applications like stereo matching, video compression, object tracking, and object segmentation. Several approaches have been proposed for its computation. Lucas and Kanade [34] tackled the aperture problem by solving for the parameters of a constant motion model over image patches. Horn and Schunck [25] were the first to use functional minimization for solving optical flow problems employing mathematical tools from calculus of variations. Their pioneering work offered the basic idea for solving dense optical flow fields for the whole image by using a functional with two terms: A data term penalizing for deviations from the optical flow equation, and a smoothness term penalizing for variations in the flow field. Several important modifications have been proposed following their work. Nagel [38, 39] proposed an oriented smoothness term that penalizes anisotropically variations in the flow field according to the direction of the intensity gradients. Replacing quadratic penalty by robust statistics integral measures was proposed
in [14, 6, 18] in order to allow sharp discontinuities in the optical flow solution along motion boundaries. Using multi-frame formulations instead of the two-frames formulation allowed to use a spatio-temporal smoothness instead of the original spatial smoothness term [8, 22, 38, 55]. Brox-Bruhn-Papenberg-Weickert [10] demonstrated the importance of using the exact optical flow equation instead of its linearized version and added gradient constancy to the data term which is important in the presence of scene illumination changes. Here, we propose to tackle the optical flow and image restoration problems in a unified approach within a variational framework. The basic idea comes from looking at the errors in the data term. These errors can be roughly classified into two main categories, errors in the computed flow field, and errors in the image itself caused by noise, optical blur, lossy compression, interlacing, etc. We propose a variational formulation of the problem that solves simultaneously for both the optical flow and the restored image sequence. In traditional optical flow computation, the images are pre-filtered. This pre-filtering is independent of the computed flow. While in image restoration, some methodologies first compute the optical flow (see for example [20]) as a pre-processing stage which is independent of the restored images.

In section 2.2 we introduce our framework for combined denoising and optical flow computation. Section 2.3, discusses parameter settings and implementation considerations. section 2.4 describes the experiments conducted to evaluate our method. Finally, section 2.5 concludes the chapter.

2.2 Problem Formulation

Given an image sequence \( I_0 : \Omega \subset \mathbb{R}^2 \times [0,T] \rightarrow [0,1] \), which is a sum of \( I^c \), an (unknown) clean image sequence, and \( n \) that represents noise, so that
at each point in space-time $I_0(x,y,t) = I^c(x,y,t) + n(x,y,t)$, we wish to find the dense optical flow field $(u(x,y,t), v(x,y,t))$ and an image sequence $I$, so that $I$ approximates $I^c$ and $I(x,y,t)$ is approximated in the best possible way by $I(x+u(x,y,t), y+v(x,y,t), t+1)$. Here we discretized $t$ to be the frame index and without loss of generality assume $dt = 1$. Again, the given noisy image sequence will be denoted by $I_0$, while $I$ will be used as an argument for our optimization procedure and represent the denoised video which we would like to push as close as possible to $I^c$.

2.2.1 Traditional optical flow functionals

Traditional optical flow functionals usually include two terms: a data term $E_D(u,v)$, that measures the deviation from the optical flow equation, and a regularization smoothness term $E_S(u,v)$ that quantifies the smoothness of the flow field. Overall, the flow field solution should minimize the sum of the data and smoothness terms.

$$E(u,v) = E_D(u,v) + E_S(u,v)$$ (2.1)

The main difference between the various variational methods is in the choice for the data and smoothness terms, and the numerical methods used for solving the minimizing flow field $(u(x,y,t), v(x,y,t))$.

2.2.2 Joint optic-flow and video restoration

The functional given in Equation (2.1) is minimized with respect to the optical flow $(u,v)$. In fact, almost all optical flow integral measures in the literature are minimized with respect to the optical flow functions $(u,v)$ alone. Let us note that there are two sources for errors in the data term,
(1) errors in the flow field, and

(2) errors in the image sequence due to noise, blur, interlacing, and lossy compression.

Writing the functional as depending only on the optical flow is appropriate mainly for ideal sequences generated by computer graphics procedures. In presence of errors in the image sequence, we should minimized with respect to the optical flow \((u, v)\) as well as the image sequence. Although, in later sections we make specific choices for the data and smoothness terms, one could employ our approach on any of the optical flow functionals suggested in the literature. The general structure of the proposed functional is

\[
E(u, v, I) = E_D(u, v, I) + E_S(u, v) + E_F(I, I_0). \tag{2.2}
\]

The functional \(E(u, v, I)\) is minimized with respect to the flow field and the image sequence. Here, \(E_D(u, v, I)\) is the gray level time-constancy data term and is treated explicitly as a function of the image sequence. \(E_S(u, v)\) is the flow field smoothness term. The additional fidelity term \(E_F(I, I_0)\) penalizes for deviations from the measured sequence. It is vital for keeping the sequence close to the given input video. Ignoring this term, one can produce many image sequences for which the data and smoothness terms both vanish, yet, have little in common with the original sequence and flow field. For example, the trivial solution \(I(x, y, t) = 0, u(x, y, t) = 0, v(x, y, t) = 0\), is a global minimum for a functional without the fidelity term.

A related approach for linking image reconstruction and optical flow using optimal control methodology has appeared in [9]. However, there, the smoothing and data terms are somewhat naive and go back to the linearized optical flow
equation with quadratic penalizers (introduced by Horn and Schunck) which is far less accurate relative to more modern functionals as shown in [10].

### 2.2.3 Specific choice for the functional

We choose to use the functional proposed in [10] excluding only the gradient constancy element from the data term. The data term is

\[
E_D(u, v) = \int \Psi((I(x + w) - I(x))^2) dx
\]  

(2.3)

Where, \( x = (x, y, t)^T \) and \( w = (u, v, 1)^T \). The integration domain is includes the space time volume of the image sequence. The function \( \Psi(s^2) = \sqrt{s^2 + \varepsilon^2} \) induces an \( L_1 \) metric on the data term as \( \varepsilon \) approaches zero.

The smoothness term is given by

\[
E_S(u, v) = \alpha \int \Psi(\|\nabla_3 u\|^2 + \|\nabla_3 v\|^2) dx.
\]  

(2.4)

Where \( \nabla_3 \) denotes the spatio-temporal gradient. Coupling these two terms, the integral measure we would like to minimize is

\[
E(u, v) = E_D(u, v) + E_S(u, v).
\]  

(2.5)

Finally, we add the fidelity term, treat the data term as a function of \( u, v, I \) and rewrite the functional as

\[
\]  

(2.6)

The data term is the same as in Equation (2.3), the only difference is that now the image sequence is treated as a variable in the minimization process. The smoothness term of the flow field remains the same and depends only on the flow. The fidelity term penalizes for deviations from the initial given noisy sequence.
We choose a quadratic term similar to the total variation denoising methodology [47],

\[ E_F(I) = \lambda \int (I - I_0)^2 d\mathbf{x}. \] (2.7)

The above fidelity term is appropriate for denoising. However, if in addition to noise, the images were degraded by a linear shift invariant filter with a given kernel \( h \), then, the fidelity term can be modified to

\[ E_F(I) = \lambda \int (h \ast I - I_0)^2 d\mathbf{x}. \] (2.8)

In this case, the restored sequence filtered by \( h \) should be close to the initial given sequence \( I_0 \) while the optical flow is calculated with respect to the restored sequence \( I \).

We combined the two integral measures,

- the optical flow computation, and
- the sequence restoration.

The optical flow module receives as an input an image sequence \( I \) and computes its \((u, v)\) flow. The sequence restoration module operates on the measured image sequence \( I_0 \) and an optical flow field \((u, v)\), and computes the restored sequence \( I \). Coupling the two modules is performed by feeding the optical flow from the first module to the second one. The resulting restored sequence is supplied as an input for the optical flow module. We iterate between these two procedures several times in order to refine the solution.

The optical flow module solves the Euler-Lagrange equations with respect to \( u \) and \( v \), that is,

\[ \Psi'(I^2_z)I_xI_z - \alpha \cdot \text{div}(\Psi'(\|\nabla_3 u\|^2 + \|\nabla_3 v\|^2)\nabla_3 u) = 0 \]
\[ \Psi'(I_z^2)I_yI_z - \alpha \cdot \text{div}(\Psi'(\|\nabla_3u\|^2 + \|\nabla_3v\|^2)\nabla_3v) = 0 \]

The numerical solution of the Euler-Lagrange equations is similar to the one described in [10], see next section for more details.

The denoising restoration scheme is derived in the following way. First, we approximate the integral by a discrete summation. Next, we use bilinear approximation in each term in the summation when required. Finally, each term in the summation is differentiated with respect to the sequence elements on which it depends. The result of the last step is used in a gradient descent scheme as an optimization process to iteratively refine the sequence volume elements.

Next, consider,

\[ \int \lambda (I - I_0)^2 + \Psi((I(x + w) - I(x))^2)dx. \]  \hspace{1cm} (2.9)

The discrete approximation of this integral is given by

\[ \sum \lambda (I - I_0)^2 + \Psi((I(x + w) - I(x))^2). \]  \hspace{1cm} (2.10)

Let us use bilinear approximation for \( I(x+w) \) in the summation of Equation (2.10). We have that,

\[ \lambda (I - I_0)^2 + \Psi((A \cdot I_1 + B \cdot I_2 + C \cdot I_3 + D \cdot I_4 - I)^2) \] \hspace{1cm} (2.11)

where,

\[
\begin{align*}
I &= I(x, y, t) \\
I_1 &= I(x_1, y_1, t + 1) \\
I_2 &= I(x_1 + 1, y_1, t + 1) \\
I_3 &= I(x_1, y_1 + 1, t + 1)
\end{align*}
\]
\[ I_4 = I(x_1 + 1, y_1 + 1, t + 1) \]
\[ x_1 = \lfloor x + u \rfloor \]
\[ y_1 = \lfloor y + v \rfloor \]

Again, note that here we assume w.l.o.g. \( dx = dy = dt = 1 \).

Differentiating Equation (2.11) with respect to each of the five image variables is straightforward. Each element in the summation of (2.10) depends on one image entry at frame \( t \) and four image entries at \( t + 1 \).

Note that each pixel value in the sequence volume may influence several optical flow paths and therefore collect several contributions from different terms in the summation of equation (2.10).

\( A, B, C, D \) are the bilinear interpolation coefficients and depend only on the optical flow at \( x, y, t \). Therefore, \( A, B, C, D \) do not change during the image restoration iterations, as the optical flow solution is numerically frozen during these iterations.

\[
A = 1 - dy - dx + dx \cdot dy; \quad B = dx - dx \cdot dy
\]
\[
C = dy - dx \cdot dy; \quad D = dx \cdot dy
\]
\[
dx = x + u - x_1; \quad dy = y + v - y_1.
\]

Note that in the restoration phase, the data term yields smoothing of the image sequence along the optical flow trajectories.

### 2.3 Implementation details

We implemented the optical flow methodology of [10], with the following exceptions.

1. We excluded the gradient constancy from the data term.
(2) In the multi-resolution we used an image reduction factor of 0.5 instead of 0.95.

(3) We used one loop of iterations instead of dividing the loops into three levels of iterations. We used 600 iterations for each optical flow calculation.

As for the denoising module, if we choose $\lambda \to \infty$, then the sequence is forced to be identical to the original one which is equivalent to all existing optic flow calculation. If we choose $\lambda = 0$, then the sequence can be modified with no penalty (equivalent to no fidelity term). As explained earlier, such a selection might produce the trivial solution to $I, v, u = 0$ as an optimal solution.

When assuming no knowledge of the noise level, we empirically found $\lambda = 0.01$ suitable for most Gaussian noise conditions. If we have some knowledge of the noise, like an approximation of the standard deviation, then $\lambda$ should decrease as the noise level increases. The sequence is thereby allowed to drift further apart from the given noisy $I_0$. we modified the value of $\lambda$ according to $\lambda \sim 1/$NoiseSTD and according to $\lambda \sim 1/$NoiseSTD$^2$. We empirically found that $\lambda \sim 1/$NoiseSTD gives better results, for $\lambda = 0.4/$NoiseSTD. In order to avoid singularities, we bounded the value of $\lambda \leq 1$.

A more academic approach to consider the noise level would be to enforce the constraint on the standard deviation of the restored sequence distance from the original to match the noise level standard deviation, and solve for the corresponding $\lambda$. Alternatives for estimating $\lambda$ based on the noise characteristics would be explored elsewhere.

We used a gradient descent on the sequence values with a numerical step size of 0.1, and 400 iterations at each denoising phase. The iterations between the optical flow and the denoising modules is performed 12 times. For the opti-
cal flow we used central derivatives and the image derivatives were computed as recommended in [4].

### 2.4 Experimental Results

In this section we compare the results of the optical flow module alone to the results of the coupled optical flow and denoising modules. In all the experiments, the optical flow code and parameters are identical in both cases and the same number of optical flow iterations is performed. We also compare the optical flow for noisy data results to the best known results in the literature. In our evaluations we adopt the standard measures of Average Angular Errors (AAE) and Standard Deviation (STD). All our results are measured on all the pixels of the flow field (100% dense).

#### 2.4.1 Yosemite sequence

In this section we applied our method to the Yosemite sequence without clouds, we used three resolution levels. Table 2.1 shows the noise sensitivity results for our optic flow code. Comparing to the results in table 2.2, we see that the optical flow estimation for noisy images improves significantly when applying the denoising part. The results reported by [10] are somewhat better than our results for the pure optic flow solution, partly due the missing gradient constancy term in our data term, and because of implementation differences from the code in [10] (which is not yet publicly available). However, our coupled solution achieves better results with respect to the AAE measure in the presence of significant noise levels, see Figure 2.1 for comparison. Figure 2.3 shows the calculated flow field of the coupled approach under heavy noise of 40. Notice that changing $\lambda$ according
to the noise level improves the results for low noise levels compared to a constant \( \lambda \). It is yet interesting to note that although the sequence is synthetic, when we changed \( \lambda \), the results improved by a small amount relative to working with the original sequence. We assume that the small amount of smoothing along the optical flow trajectories reduced the effects of the rendering algorithm used in producing the sequence. This result is also valid for other sequences we tried like the office sequence. Figure 2.2 shows the denoising results of the coupled solution, the noise standard deviation is reduced from 40 in the original sequence to 18 in the denoised sequence. Note that the denoising is less effective at boundary pixels at the lower left/right image since the optical flow there is fast and moves out of the image plane. Therefore, there are fewer pixels to work with in the smoothing process.

Table 2.1: Yosemite without clouds. Optical flow alone.

<table>
<thead>
<tr>
<th>( \sigma_n )</th>
<th>AAE [deg]</th>
<th>STD [deg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.22°</td>
<td>1.25°</td>
</tr>
<tr>
<td>10</td>
<td>1.60°</td>
<td>1.50°</td>
</tr>
<tr>
<td>20</td>
<td>2.02°</td>
<td>1.73°</td>
</tr>
<tr>
<td>30</td>
<td>2.45°</td>
<td>2.01°</td>
</tr>
<tr>
<td>40</td>
<td>2.92°</td>
<td>2.27°</td>
</tr>
</tbody>
</table>

Table 2.2: Yosemite without clouds. Coupled solution of optical flow and sequence denoising. Left: \( \lambda = 0.4/\text{NoiseStandardDeviation} \). Right: constant \( \lambda = 0.01 \)

<table>
<thead>
<tr>
<th>( \sigma_n )</th>
<th>AAE [deg] ± STD [deg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.20 ± 1.25, 1.42 ± 1.50</td>
</tr>
<tr>
<td>10</td>
<td>1.29 ± 1.38, 1.44 ± 1.53</td>
</tr>
<tr>
<td>20</td>
<td>1.55 ± 1.51, 1.57 ± 1.56</td>
</tr>
<tr>
<td>30</td>
<td>1.86 ± 1.69, 1.85 ± 1.69</td>
</tr>
<tr>
<td>40</td>
<td>2.17 ± 1.88, 2.17 ± 1.88</td>
</tr>
</tbody>
</table>
Table 2.3: Yosemite without clouds. Results reported in [10].

<table>
<thead>
<tr>
<th>$\sigma_n$</th>
<th>AAE</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$0.98^\circ$</td>
<td>$1.17^\circ$</td>
</tr>
<tr>
<td>10</td>
<td>$1.26^\circ$</td>
<td>$1.29^\circ$</td>
</tr>
<tr>
<td>20</td>
<td>$1.63^\circ$</td>
<td>$1.39^\circ$</td>
</tr>
<tr>
<td>30</td>
<td>$2.03^\circ$</td>
<td>$1.53^\circ$</td>
</tr>
<tr>
<td>40</td>
<td>$2.40^\circ$</td>
<td>$1.71^\circ$</td>
</tr>
</tbody>
</table>

Table 2.4: Yosemite without clouds. Results reported in [12].

<table>
<thead>
<tr>
<th>$\sigma_n$</th>
<th>AAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$1.79^\circ$</td>
</tr>
<tr>
<td>10</td>
<td>$2.53^\circ$</td>
</tr>
<tr>
<td>20</td>
<td>$3.47^\circ$</td>
</tr>
<tr>
<td>40</td>
<td>$5.34^\circ$</td>
</tr>
</tbody>
</table>

Figure 2.1: Yosemite sequence - noise sensitivity results.
Figure 2.2: Frame 8. Left-original. Middle-image with noise STD of 40. Right-denoised image.

Figure 2.3: Optical flow of the Yosemite sequence (excluding the sky region). Left - Ground truth. Right - results obtained by the coupled solution with noise standard deviation of 40.
2.4.2 Office sequence

Next we applied our method to the office sequence. For this, we used the coupled modules with varying $\lambda$ and four resolution levels. Table 2.5 shows a significant improvement achieved by the whole scheme relative to the optic flow solution as a stand alone procedure. The result is 46% better in AAE for noise level 40. The improvement here is more significant than in the Yosemite sequence for two reasons: It contains more frames (higher temporal sampling rate) which provides the denoising algorithm a better support while smoothing along the optic flow trajectories. Due to the same reason, the motion field in the Yosemite sequence is larger and therefore some optic flow trajectories are shorter because they point out of the image domain. Our results for noisy sequences are significantly better than those reported in the literature (see table 2.6). Under heavy noise (STD = 40) our results are 78% better than the best result reported in the literature for this sequence. Figure 2.4 shows the denoising results obtained by our method. The noise STD in the image is reduced from 40 to 17.

Table 2.5: Our results on the office sequence. Left - coupled solution. Right - optic flow alone.

<table>
<thead>
<tr>
<th>$\sigma_n$</th>
<th>AAE</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3.24°, 3.25°</td>
<td>3.79°, 3.79°</td>
</tr>
<tr>
<td>10</td>
<td>3.63°, 4.08°</td>
<td>3.95°, 4.14°</td>
</tr>
<tr>
<td>20</td>
<td>4.09°, 5.01°</td>
<td>4.09°, 4.52°</td>
</tr>
<tr>
<td>30</td>
<td>4.82°, 6.51°</td>
<td>4.27°, 5.13°</td>
</tr>
<tr>
<td>40</td>
<td>6.04°, 8.81°</td>
<td>4.89°, 6.48°</td>
</tr>
</tbody>
</table>
Table 2.6: Office sequence. AAE Results from the literature LK=Lucas and Kanade. HS=Horn and Schunck. CLG= Combined local global approach [12].

<table>
<thead>
<tr>
<th>$\sigma_n$</th>
<th>LK</th>
<th>HS</th>
<th>CLG-2D</th>
<th>CLG-3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5.75$^\circ$</td>
<td>4.36$^\circ$</td>
<td>4.32$^\circ$</td>
<td>3.24$^\circ$</td>
</tr>
<tr>
<td>10</td>
<td>6.79$^\circ$</td>
<td>6.17$^\circ$</td>
<td>5.89$^\circ$</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>8.43$^\circ$</td>
<td>8.30$^\circ$</td>
<td>7.75$^\circ$</td>
<td>-</td>
</tr>
<tr>
<td>40</td>
<td>11.47$^\circ$</td>
<td>11.76$^\circ$</td>
<td>10.73$^\circ$</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 2.4: Frame 10 of the office sequence. From left to right: Original, with noise STD=40, and Denoised.
2.5 Summary

In this work we introduced the concept of simultaneously solving the optical flow and the video denoising by minimizing a single functional. The results demonstrated a significant improvement of the optical flow results under noise compared to optical flow implementation without the coupled image denoising. The optical flow provides a vital information for the denoising algorithm: The knowledge about trajectories along which the image brightness should be constant. While the denoising part provides the optical flow with an improved image sequence with lower noise levels. It might be interesting in future work to test different functionals. For example, one may incorporate the gradient constancy into the data term. It would change both the optical flow as well as the denoising scheme. That is, a denoising that also tries to improve the match of the image gradients along the optical flow trajectories.
Chapter 3

Model Based Total Variation Regularization

3.1 Introduction

The Total Variation (TV) approach for image denoising was first proposed by Osher-Rudin-Fatemi [47]. In the TV approach, we seek for a minimizer of a functional by a variational approach, where, the regularization term in the functional measures the smoothness of the solved function by an $L_1$ measure. The TV regularization approach is relevant not only to images but also to general (one or higher dimensional) function regularization, or in other problems, such as optical flow computation. For example, Papenberg-Bruhn-Brox-Didas-Weickert [45] proposed to estimate the optical flow using $L_1$ regularization in both the data and smoothness terms in their variational formulation. The advantages of an $L_1$ regularizer resulted in many state of the art variational methods. Such a regularizer is defined on the border between convex and non-convex functions, it is able to preserve sharp discontinuities, and it is less sensitive to outliers than its $L_2$ alternative. The main disadvantage of TV or $L_1$-regularization is that it complicates the numerical scheme and requires more computation time and some small $\epsilon$ convexification. We here discuss a novel approach to regularization which relies on over-parameterizing the space of functions over which we seek a
variational solution.

3.2 Problem formulation

Let us start, for pedagogical reasons, with functions in one dimension. Suppose we have an ideal continuous function $f_{\text{ideal}}(x)$ that we wish to estimate from its noisy samples: $f_{\text{Noisy}}(x) = f_{\text{ideal}}(x) + n(x)$ where, $n(x)$ is zero mean white Gaussian noise $n(x) \sim N(0, \sigma_{\text{Noise}})$. Let us define the following functional

$$E(f) = \int (f(x) - f_{\text{Noisy}}(x))^2 dx + \alpha \int (f'(x))^2 dx.$$  \hspace{1cm} (3.1)

The first term (data/fidelity term) requires that the solution would be close to the measurement and the second term (regularization or smoothness term) is a data independent regularization term which enforces a smooth solution. The resulting Euler-Lagrange equation is

$$f(x) - f_{\text{Noisy}}(x) - \alpha f''(x) = 0.$$  \hspace{1cm} (3.2)

The constant $\alpha$ is the relative weight of the regularization. For $\alpha \to 0$ we obtain the trivial solution of $f(x) = f_{\text{Noisy}}$. For $\alpha \to \infty$ we obtain the solution $f''(x) = 0$ from the Euler-Lagrange equation, which results in a linear solution for $f(x)$. However, Equation (3.1) shows that only the constant solution produces a finite penalty of the functional. Therefore, in this case the resulting solution is constant and by observing the data term, the value of this constant is the average of $f_{\text{Noisy}}$ over the integration interval. The solution of the EL equations can be numerically sought after via a gradient descent algorithm which initializes $f(x) = f_{\text{Noisy}}(x)$ and then performs gradient descent iterations of the form $f_{k+1}(x) = f_k(x) - \Delta t(f_k(x) - f_{\text{Noisy}}(x) - \alpha f''_k(x))$. 
Let us now replace the $L_2$ regularizer by a TV $L_1$ regularizer of the form

$$E(f) = \int (f(x) - f_{\text{Noisy}}(x))^2 dx + \alpha \int \sqrt{f'(x)^2 + \varepsilon^2} dx.$$ \hspace{1cm} (3.3)

The resulting Euler-Lagrange equation

$$2(f(x) - f_{\text{Noisy}}(x)) - \alpha \frac{d}{dx} \left( \frac{f'(x)}{\sqrt{f'(x)^2 + \varepsilon^2}} \right) = 0.$$ \hspace{1cm} (3.4)

After a few algebraic manipulations we obtain

$$2(f(x) - f_{\text{Noisy}}(x)) - \alpha \varepsilon^2 f''(x) \left( \frac{f'(x)^2 + 2\varepsilon^2}{(f'(x)^2 + \varepsilon^2)^{1.5}} \right) = 0.$$ \hspace{1cm} (3.5)

One could also replace the $L_2$ norm in the data term with an $L_1$ norm of equation (3.3). In this case, we obtain the functional

$$E(f) = \int \sqrt{(f(x) - f_{\text{Noisy}}(x))^2 + \varepsilon^2} dx + \alpha \int \sqrt{f'(x)^2 + \varepsilon^2} dx,$$ \hspace{1cm} (3.6)

that yields the Euler-Lagrange,

$$\frac{f(x) - f_{\text{Noisy}}(x)}{\sqrt{(f(x) - f_{\text{Noisy}}(x))^2 + \varepsilon^2}} - \alpha \frac{d}{dx} \left( \frac{f'(x)}{\sqrt{f'(x)^2 + \varepsilon^2}} \right) = 0.$$ \hspace{1cm} (3.7)

Figure 3.1 shows a piecewise constant function with one discontinuity and its noisy sample. Using an $L_2$ regularizer results in reduction of the noise influence while smoothing of the discontinuity of the original signal. If we replace the regularizer with the TV $L_1$ norm of Equation (3.3), we obtain the result shown in Figure 3.2 where the discontinuity is well preserved.

Let us now examine the effect of an outlier when using an $L_1$ versus $L_2$ norm in the data term. The noisy signal of example 1 excluding the outlier is shown in Figure 3.1, one outlier point is now placed at $x = 100$ with the measurement $y_{\text{Noisy}}(100) = -3$. The influence on the error graph is shown in Figure 3.3. Placing an $L_1$ norm in the data term has the effect shown in the error graph of Figure 3.4, where the influence of the outlier is very small.
Figure 3.1: Example 1 with $L_2$ data term and $L_2$ regularization.

Figure 3.2: Example 1 with $L_2$ data term and $L_1$ regularization.
Figure 3.3: Error graph of example 1 with $L_2$ data term and $L_1$ regularization, showing the influence of a single outlier.
Figure 3.4: Error graph of example 1 with $L_1$ data term and $L_1$ regularization, showing the negligible influence of a single outlier in this case.
3.3 Total variation of piecewise parametric functions

Let us analyze the case of a clean signal with $f_{\text{Noisy}}(x) = f_{\text{Ideal}}(x)$, and ask ourselves: what types of functions can the TV denoising functional recover faithfully, i.e. recover without an error? In order for any function to be represented without an error, disregarding the noise, all we have to do is to substitute $f(x) = f_{\text{Ideal}}(x)$ in the corresponding Euler-Lagrange equation and check if it is satisfied. In this substitution, the data term vanishes and all we have to do is check the term corresponding to the regularizer. Looking at equations (3.2) and (3.5), we observe that for both the $L_1$ and $L_2$ regularization methods, the Euler-Lagrange equation is satisfied for $f(x) = f_{\text{Ideal}}(x) = f_{\text{Noisy}}(x)$ if and only if $f''_{\text{Ideal}}(x) = 0$ everywhere. The solution of this simple differential equation leads to a linear function. We conclude that the previously presented denoising functionals can faithfully reproduce linear functions. From a practical point of view, even if the ideal function can not be exactly represented, this does not necessarily mean that the functional is inadequate. We have to keep in mind that due to noise, the restored function would generally be only an approximation to the ideal function, and therefore, practically we would settle for errors in the noise free case that are an order of magnitude less than the errors resulting from the contribution of the noise. In the next section we will see how knowledge of the nature of the ideal function can be incorporated into the TV functional in order to increase the accuracy of the restoration.
3.4 An over-parameterized model for the 1D total variation functional

Suppose next that we know that our ideal function is composed of a linear combination of \( n \) known "basis functions" \( \phi_i(x) \)

\[
f_{\text{Ideal}}(x) = \sum_{i=1}^{n} A_i \phi_i(x).
\]  

(3.8)

In this case, the problem reduces to parameter estimation, and the use of functional minimization is unnecessary. Suppose however that we know that the ideal function may be represented with the help of basis functions \( \phi_i(x) \) with varying coefficients \( A_i(x) \), i.e.

\[
f_{\text{Ideal}}(x) = \sum_{i=1}^{n} A_i(x) \phi_i(x),
\]  

(3.9)

For most choices of basis functions, we do not impose here any restrictions on the representation possibilities of any \( f_{\text{Ideal}}(x) \). For more than one basis function, the representation is not unique and there may be infinite ways to represent the same \( f_{\text{Ideal}}(x) \) function. Suppose for example that \( \phi_0(x) = 1 \). In this case, \( A_0(x) = f_{\text{Ideal}}(x) \) and \( A_i(x) = 0 \) for all \( i > 0 \) is one possibility to represent any ideal function. If basis function number \( k \), \( \phi_k(x) \) is none zero anywhere, then \( A_k(x) = f_{\text{Ideal}}(x) / \phi_k(x) \) with all the other coefficients equal to zero, is another possible exact representation of \( f_{\text{Ideal}}(x) \). If the coefficient functions are piecewise constant, we can still try the approach of the parameter estimation, but now one also has to solve a segmentation problem of deciding on the locations of segments that have constant coefficients. In the noise free case simultaneous segmentation and parameter estimation may produce excellent results. In the noisy case on the other hand, the problem is much more difficult. The solution we propose is to
incorporate the model of Equation (3.10) into the functional optimization, so that we now solve for \( f(x) \) which is represented by

\[
f(x) = \sum_{i=1}^{n} A_i(x)\phi_i(x).
\] (3.10)

Now, the regularization is expressed in terms of the coefficients \( A_i(x) \) instead of \( f(x) \). Also, the Euler-Lagrange equations are written in terms of the coefficients instead of \( f(x) \). For an \( L_2 \) functional we would use

\[
E(A_i) = \int \left( \sum_{i=1}^{n} A_i(x)\phi_i(x) - f_{Noisy}(x) \right)^2 dx + \alpha \int \sum A_i'(x)^2 dx.
\] (3.11)

For an \( L_1 \) regularizer and an \( L_2 \) data term we get

\[
E(A_i) = \int \left( \sum_{i=1}^{n} A_i(x)\phi_i(x) - f_{Noisy}(x) \right)^2 dx + \alpha \int \sqrt{\sum_{i=1}^{n} A_i'(x)^2 + \varepsilon^2 dx}
\] (3.12)

This is the over-parameterized generalization of the classical total variation functional in 1D. Note that for piecewise constant coefficients of an ideal function in Equation (3.9), the solution \( f(x) \) can model the ideal solution with no regularization penalty except at the points of discontinuity in the model coefficients.

For the sake of simplicity we avoided writing explicit weights for the derivatives of the different coefficients in the above functional since we choose equivalently to scale the basis functions. Since we now minimize the functional with respect to the coefficients, there are now \( n \) Euler-Lagrange equations which can be written for the most general choice of basis functions. The Euler-Lagrange equation for coefficient number \( q \) is

\[
2 \left( \sum_{i=1}^{n} A_i(x)\phi_i(x) - f_{Noisy}(x) \right) \phi_q - \alpha A_q''(x) = 0
\] (3.13)
for the functional of Equation (3.11), and

\[ 2 \left( \sum_{i=1}^{n} A_i(x) \phi_i(x) - f_{\text{Noisy}}(x) \right) \phi_q - \alpha \frac{d}{dx} \left( \frac{A_q'(x)}{\sqrt{\sum_{i=1}^{n} A_i'(x)^2 + \varepsilon^2}} \right) = 0 \quad (3.14) \]

for the functional of Equation (3.12). There are \( n \) such Euler-Lagrange equations for every \( q = 1 \ldots n \). Note that for the choice of one basis function \( \phi_1(x) = 1 \), the coefficient of this basis function becomes the function itself \( f(x) = A_1(x) \) and the Euler-Lagrange Equation (3.14) reduces to Equation (3.4). This shows that in some sense, our method may be viewed as a generalization of the total variation framework in conjunction with the constant basis function model. Without noise, a perfect reconstruction is achieved for the case of an ideal function with linearly varying coefficients as can be seen by substitution of the ideal function into the Euler-Lagrange Equations (3.13) and (3.14). The new regularization strategy obtained by the over-parameterization functional penalizes for changes in the underlying model coefficients instead of penalizing for any change in the function itself. In example 2, consider the denoising of a piecewise quadratic function with two regions of different quadratic coefficients, shown in Figure 3.5. Since according to our previous discussion, if the model coefficients are allowed to change linearly and the basis functions are also linear, the overall space of exact representation of noise free ideal functions would be quadratic. The classical and over-parameterized regularization reconstruction errors for the noise free case are shown in Figure 3.6. Adding noise with standard deviation of 0.01 results in the errors shown in Figure 3.7, which shows that for low noise levels, the approximation error of the classical total variation method is higher than the influence of the noise on the over-parameterized model. For 5 times larger noise standard deviation, the errors for the two methods are shown in Figure 3.8. Since the noise sensitivity of the over-parameterized formulation is larger than the sensitivity of
the regular model, the influence of the representation error which appears in the classical TV is of about the same order of magnitude as the influence of the noise on the over-parameterized model.

3.4.1 Implementation details for the 1D examples

In all the examples, we assumed a sampling interval of $dx = 1$. We used central first and second derivatives and reflecting boundary conditions. In all the methods, we used $\alpha = 80$. The value of the parameter $\varepsilon$ in the $L_1$ functionals was chosen to be $\varepsilon = 0.1$. The solution method used was gradient descent with 8000 iterations. The linear basis functions for the over-parameterized method where: $\phi_1 = 1$ and $\phi_2 = 0.4(x - 200.5)$, where, a scaling factor of 0.4 was empirically chosen, and 200.5 is the midpoint of the span of the $x$ values which range from 1 to 400.

3.5 Over-Parameterized image representations for total variation image denoising

The derivation of the results from denoising of $1D$ functions to $2D$ functions is straightforward. The classical total variation for $2D$ functions (images) as presented in [47] has an $L_2$ measure for the data term and an $L_1$ measure for the regularizer

$$E(f) = \int \int (f(x, y) - f_{Noisy}(x, y))^2dxdy +$$

$$\alpha \int \int \sqrt{f_{x}(x, y)^2 + f_{y}(x, y)^2 + \varepsilon^2}dxdy$$
Figure 3.5: Piecewise quadratic function of example 2.
Figure 3.6: Example 2: reconstruction errors for the regular (dotted) versus linear over-parameterized total variation for the noise free case.
Figure 3.7: Example 2: reconstruction errors for the regular (dotted) versus linear over-parameterized total variation for noise STD=0.01.
Figure 3.8: Example 2: reconstruction errors for the regular (dotted) versus linear over-parameterized total variation for noise STD=0.05.
The resulting Euler-Lagrange equation

\[
2(f(x, y) - f_{\text{Noisy}}(x, y)) - \alpha \frac{d}{dx} \left( \frac{f_x(x, y)}{\sqrt{f_x(x, y)^2 + f_y(x, y)^2 + \epsilon^2}} \right) - \alpha \frac{d}{dy} \left( \frac{f_y(x, y)}{\sqrt{f_x(x, y)^2 + f_y(x, y)^2 + \epsilon^2}} \right) = 0
\]  

(3.16)

The last equation can be written more compactly using the div operator

\[
2(f(x, y) - f_{\text{Noisy}}(x, y)) - \alpha \cdot \text{div} \left( \frac{\nabla f(x, y)}{\sqrt{f_x(x, y)^2 + f_y(x, y)^2 + \epsilon^2}} \right) = 0
\]

(3.17)

The over-parameterized representation of the 2D function \( f(x, y) \) is

\[
f(x, y) = \sum_{i=1}^{n} A_i(x, y) \phi_i(x, y),
\]

(3.18)

where now, both the coefficients and basis functions are 2D functions. The over-parameterized generalization of the functional of Equation (3.15) is

\[
E(f) = \int \int \left( \sum_{i=1}^{n} A_i(x, y) \phi_i(x, y) - f_{\text{Noisy}}(x, y) \right)^2 dxdy + \alpha \int \int \sqrt{\sum_{i=1}^{n} A_{ix}(x, y)^2 + A_{iy}(x, y)^2 + \epsilon^2} dxdy
\]

(3.19)

where, \( A_{ix} \) and \( A_{iy} \) denotes the partial derivative of \( A_i \) with respect to the \( x \) and \( y \) axes respectively. The \( n \) Euler-Lagrange equations corresponding to the functional of Equation (3.19) are obtained for \( q = 1...n \)

\[
E(f) = \left( \sum_{i=1}^{n} A_i(x, y) \phi_i(x, y) - f_{\text{Noisy}}(x, y) \right) \phi_q(x, y) - \alpha \cdot \text{div} \left( \frac{\nabla A_q(x, y)}{\sqrt{\sum_{i=1}^{n} A_{ix}(x, y)^2 + A_{iy}(x, y)^2 + \epsilon^2}} \right).
\]

(3.20)

Here too, as can be observed from Equation (3.20), the terms resulting from the regularization are all multiplied by a second derivative (\( \partial_{xx}, \partial_{yy} \) or \( \partial_{xy} \)) of some coefficient (as can be seen after expansion of the div operator), therefore,
as in the 1D case, the Euler-Lagrange equations are satisfied for any noise free ideal function which can be described by linearly varying coefficients (of the form: $A_i = C_1 + C_2 x + C_3 y$) multiplying the set of basis functions of our choice.

### 3.5.0.1 Prisoner’s room example

Let us now examine a synthetic example: suppose a prisoner in a jail is secretly taking photographs of his cell, if the pictures were taken with a flashlight, they would have looked something like the left image in Figure 3.9, but since the room was quite dark (right image of Figure 3.9) and using the flash would be too dangerous, the photo came out dark. As the prisoner finally finished serving his sentence of imprisonment, he went to see a friend who studied computer vision who wanted to help him get a better image of his cell. First he stretched the gray level values by uniform scaling. Unfortunately, the noise became also more visible (inherent signal to noise problem) as shown on Figure 3.10 left, so he tried the total variation denoising approach with the results shown on the middle image of Figure 3.10, where the bars are blurred. He then tried the over-parameterized total variation approach using the information the prisoner (who happened to be a smart engineer) gave him: the light changed linearly on every plane of the room and the light falling on the bars on the window formed a harmonic pattern in the $x$ direction of the image with frequency of $f = 0.25[1/pixel]$. Using this information his friend constructed 3 basis functions: $\phi_1 = 1$, $\phi_2 = sin(2\pi fx)$ and $\phi_3 = cos(2\pi fx)$. The image and error using this method are shown in Figure 3.11. As can be seen the pattern of the bar is much better reconstructed using the proper basis functions. In terms of signal to noise ratio, the noisy image has a $PSNR = 33.96\, dB$, the regular total variation has $PSNR = 28.05\, dB$ (due
mainly to errors in the region of the bars), the over-parameterized model has $PSNR = 35.79dB$, all compared to the noise free image.

Figure 3.9: Prisoner’s room. Left - Ideal image. Right - Dark photo with noise STD=5.1.

3.6 Summary

In this chapter, we have seen how the total variation approach can be generalized to incorporate a model of choice. The model is used by assigning each sample/pixel its own independent set of coefficients. The regularization applied to the derivatives of the coefficients penalizes for deviations from the model parameters. Conditions for having zero representation errors for the noise free case where derived. The next chapter examines the application of this methodology to the optical flow computation problem.
Figure 3.10: Prisoner’s room. Left - Stretched image. Middle - Image after total variation denoising. Right - Image after total variation denoising with the over-parameterized model.

Figure 3.11: Error image (the same linear scaling from errors to gray-levels was used for both images. Left - Errors for the regular total variation. Right - Errors for the over-parameterized total variation.
Chapter 4

Over-Parameterized Variational Optical Flow

4.1 Introduction

Despite much research effort invested in addressing optical flow computation it remains a challenging task in the field of computer vision. It is a necessary step in various applications like stereo matching, video compression, object tracking, depth reconstruction and motion based segmentation. Hence, many approaches have been proposed for optical flow computation. Most methods assume brightness constancy and introduce additional assumptions on the optical flow in order to deal with the inherent aperture problem. Lucas and Kanade [34] tackled the aperture problem by solving for the parameters of a constant motion model over image patches. Subsequently, Irani-Rousso-Peleg [27, 28] used motion models in a region in conjunction with Lucas-Kanade in order to recover the camera ego-motion. Spline based motion models were suggested by Szeliski and Coughlan [50].

Horn and Schunck [25] sought to recover smooth flow fields and were the first to use functional minimization for solving optical flow problems employing mathematical tools from calculus of variations. Their pioneering work put forth the basic idea for solving dense optical flow fields over the whole image by introducing
a quality functional with two terms: a data term penalizing for deviations from the brightness constancy equation, and a smoothness term penalizing for variations in the flow field. Several important improvements have been proposed following their work. Nagel [38, 39] proposed an oriented smoothness term that penalizes anisotropically for variations in the flow field according to the direction of the intensity gradients. Ben-Ari and Sochen [3] recently used a functional with two alignment terms composed of the flow and image gradients. Replacing quadratic penalty terms by robust statistics integral measures was proposed in [14, 6, 18] in order to allow sharp discontinuities in the optical flow solution along motion boundaries. Extensions to multi-frame formulations of the initial two-frames formulation allowed the consideration of spatio-temporal smoothness to replace the original spatial smoothness term [8, 22, 38, 55]. Brox-Bruhn-Papenberg-Weickert [10, 45] demonstrated the importance of using the exact brightness constancy equation instead of its linearized version and added a gradient constancy to the data term which is important if the scene illumination changes in time. Recently, Amiaz and Kiryati [2] followed by Brox et al. [11] introduced a variational approach for joint optical flow computation and motion segmentation. In Farneback [21, 22], a constant and affine motion model is employed. The motion model is assumed to act on a region, and optic flow based segmentation is performed by a region growing algorithm. An affine model based variational approach was proposed by Dufaux et al. [19]. In a classical contribution to structure from motion Adiv [1] used optical flow in order to determine motion and structure of several rigid objects moving in the scene. Sekkati and Mitiche [48] used joint segmentation and optical flow estimation in conjunction with a single rigid motion in each segmented region. Vazquez-Mitiche-Laganiere [54] used joint multi-region
segmentation with high order DCT basis functions representing the optical flow in each segmented region. Cremers and Soatto [15] propose a motion competition algorithm for parametric motion segmentation.

In this chapter, we propose to represent the optical flow vector at each pixel by different coefficients of the same motion model in a variational framework. Such a grossly over-parameterized representation has the advantage that the smoothness term may now penalize deviations from the motion model instead of directly penalizing the change of the flow. For example, in an affine motion model, if the flow in a region can be accurately represented by an affine model, then in this region there will be no flow regularization penalty, while in the usual setting there is a cost resulting from the changes in the flow induced by the affine model. This over-parameterized model thereby offers a richer means for optical flow representation. For segmentation purposes, the over-parametrization has the benefit of making segmentation decisions in a more appropriate space (e.g. the parameters of the affine flow) rather than in a simple constant motion model space. The work of Ju-Black-Jepson [32] is related to our methodology, they used local affine models to describe the motion in image regions imposing spatial smoothness on the affine parameters between neighboring patches. The key and conceptually very important difference being that in our approach the model is represented at the pixel level which makes the problem over-parameterized while the patch size chosen in [32] makes it under-parameterized and requires the choice of a neighborhood size.

Section 4.2 introduces the over-parameterized optical flow representation model and the corresponding functional together with the resulting Euler-Lagrange equations, examples of specific optical flow models are described. Section 4.3 discusses numerical solution considerations. Section 4.4 describes the parameter set-
tings and the experiments conducted to evaluate our method. Finally, Section 4.5 concludes the chapter.

### 4.2 Over-parametrization model

We propose to represent the optical flow \((u(x, y, t), v(x, y, t))\) by the general over-parameterized space-time model

\[
u(x, y, t) = \sum_{i=1}^{n} A_i(x, y, t)\phi_i(x, y, t)
\]

\[
v(x, y, t) = \sum_{i=1}^{n} A_i(x, y, t)\eta_i(x, y, t),
\]

where, \(\phi_i(x, y, t)\) and \(\eta_i(x, y, t), i = 1 \ldots n\) are \(n\) basis functions of the flow model, while the \(A_i\) are space and time varying coefficients of the model. This is an obviously heavily over-parameterized model since for more than two basis functions, there are typically many ways to express the same flow at any specific location. This redundancy however will be adequately resolved by a regularization assumption applied to the coefficients of the model. The coefficients and basis functions may be general functions of space-time, however, they play different roles in the functional minimization process: The basis functions are fixed and selected a priori. The coefficients are the unknown functions we solve for in the optical flow estimation process. In our model, appropriate basis functions are such that the true flow could be described by approximately piecewise constant coefficients, so that most of the local spatio-temporal changes of the flow are induced by changes in the basis functions and not by variations of the coefficients. This way, regularization applied to the coefficients (as will be described later on) becomes meaningful since major parts in the optic flow variations can be described without changes of the coefficients. For example, rigid body motion has a specific
optical flow structure which can explain the flow using only six parameters at locations with approximately constant depth. Let us start from conventional optical flow functionals that includes a data term $E_D(u, v)$, that measures the deviation from the brightness constancy assumption, and a regularization (or smoothness) term $E_S(u, v)$ that quantifies the smoothness of the flow field. The solution is the minimizer of the sum of the data and smoothness terms

$$E(u, v) = E_D(u, v) + \alpha E_S(u, v).$$ (4.2)

The main difference between the diverse variational methods is in the choice of data and smoothness terms, and in the numerical methods used for solving for the minimizing flow field $(u(x, y, t), v(x, y, t))$. For the data term we shall use the functional

$$E_D(u, v) = \int \Psi((I(x + w) - I(x))^2) dx,$$ (4.3)

where, $x = (x, y, t)^T$ and $w = (u, v, 1)^T$. This is the integral measure used for example in [10] (omitting the gradient constancy term). The function $\Psi(s^2) = \sqrt{s^2 + \varepsilon^2}$ that is by now widely used, induces an approximate $L_1$ metric of the data term for a small $\varepsilon$. The smoothness term used in [10] is given by

$$E_S(u, v) = \int \Psi \left( \|\tilde{\nabla} u\|^2 + \|\tilde{\nabla} v\|^2 \right) dx.$$ (4.4)

where $\tilde{\nabla} f \equiv (f_x, f_y, \omega_t f_t)^T$ denotes the weighted spatio-temporal gradient. $\omega_t$ indicates the weight of the temporal axis relative to the spatial axes in the context of the smoothness term ($\omega_t = 1$ is used in [10]). Inserting the over-parameterized model into the data term, we have

$$E_D(A_i) = \int \Psi \left( \left( I \left( x + \sum_{i=1}^n A_i \phi_i, y + \sum_{i=1}^n A_i \eta_i, t + 1 \right) - I(x, y, t) \right)^2 \right) dx$$ (4.5)
Our proposed smoothness term replaces equation (4.4) with a penalty for spatio-temporal changes in the coefficient functions,

$$E_S(A_i) = \int \Psi \left( \sum_{i=1}^{n} \| \tilde{\nabla} A_i \|^2 \right) \, dx. \quad (4.6)$$

Notice that in Equation (4.6), constant parameters of the model can describe changes of the flow field according to the chosen model as described in Equation (4.1) (e.g. Euclidean, affine, etc.) without smoothness penalty, whereas in Equation (4.4), any change in the flow field is penalized by the smoothness term. For the sake of simplicity of the resulting Euler-Lagrange equations, we have omitted writing explicit relative weights to the different coefficients in the smoothness term, the weighting is alternatively achieved by scaling the basis functions by appropriate factors as will be shown in the description of the motion models. Scaling a basis function by a small factor would mean that in order to achieve the same overall influence on the optical flow, the corresponding coefficient would have to make larger changes (proportional to the inverse of the factor). These larger changes would be suppressed by the regularization term. On the other hand, scaling a basis function by a large factor would scale down the changes required from the corresponding coefficient in order to achieve the same overall change and therefore would result with less regularization for this specific term.

4.2.1 Euler-Lagrange equations

For an over-parametrization model with $n$ coefficients, there are $n$ Euler-Lagrange equations, one for each coefficient. The Euler-Lagrange equation for $A_q$ ($q = 1, \ldots, n$) is given by

$$\Psi' \left( f_z^2 \right) I_z (I_x \phi_q + I_y \eta_q) - \alpha \cdot \text{div} \left( \Psi' \left( \sum_{i=1}^{n} \| \tilde{\nabla} A_i \|^2 \right) \tilde{\nabla} A_q \right) = 0. \quad (4.7)$$
where $\nabla f \equiv (f_x, f_y, \omega^2 f_t)^T$, and

$$I_x^+ := I_x \left( x + \sum_{i=1}^{n} A_i \phi_i, y + \sum_{i=1}^{n} A_i \eta_i, t + 1 \right)$$

$$I_y^+ := I_y \left( x + \sum_{i=1}^{n} A_i \phi_i, y + \sum_{i=1}^{n} A_i \eta_i, t + 1 \right)$$

$$I_z := I \left( x + \sum_{i=1}^{n} A_i \phi_i, y + \sum_{i=1}^{n} A_i \eta_i, t + 1 \right) - I(x, y, t). \quad (4.8)$$

### 4.2.2 The affine over-parametrization model

The affine model is a good approximation of the flow in large regions in many real world scenarios. We therefore first start with the affine model for our method. Note that we do not force the affine model over image patches as in previously considered image registration techniques, and here each pixel has "its own" independent affine model parameters. The affine model has $n = 6$ parameters,

$$\phi_1 = 1; \quad \phi_2 = \hat{x}; \quad \phi_3 = \hat{y};$$

$$\phi_4 = 0; \quad \phi_5 = 0; \quad \phi_6 = 0;$$

$$\eta_1 = 0; \quad \eta_2 = 0; \quad \eta_3 = 0;$$

$$\eta_4 = 1; \quad \eta_5 = \hat{x}; \quad \eta_6 = \hat{y}, \quad (4.9)$$

where,

$$\hat{x} = \frac{\rho (x - x_0)}{x_0}$$

$$\hat{y} = \frac{\rho (y - y_0)}{y_0} \quad (4.10)$$

and $x_0$ and $y_0$ are half image width and height respectively. $\rho$ is a constant that has no meaning in an unconstrained optimization such as the Lucas-Kanade
method. In our variational formulation, $\rho$ is a parameter which weighs the penalty of the $x$ and $y$ coefficients relative to the coefficient of the constant term in the regularization. An equivalent alternative is to add a different weight for each coefficient in Equation (4.6).

4.2.3 The rigid motion model

The optic flow of an object moving in rigid motion or of a static scene with a moving camera is described by

$$
\begin{align*}
    u &= -\theta_1 + \theta_3 \hat{x} + \Omega_1 \hat{x} \hat{y} - \Omega_2 (1 + \hat{x}^2) + \Omega_3 \hat{y} \\
    v &= -\theta_2 + \theta_3 \hat{y} + \Omega_1 (1 + \hat{y}^2) - \Omega_2 \hat{x} \hat{y} - \Omega_3 \hat{x}
\end{align*}
$$

(4.11)

where, $(\theta_1, \theta_2, \theta_3)^T$ is the translation vector divided by the depth and $(\Omega_1, \Omega_2, \Omega_3)^T$ is the rotation vector. Here too, the number of coefficients is $n = 6$. The coefficients $A_i$ represent the translation and rotation variables, $A_1 = \theta_1$; $A_2 = \theta_2$; $A_3 = \theta_3$; $A_4 = \Omega_1$; $A_5 = \Omega_2$; $A_6 = \Omega_3$, and the basis functions are $\phi_1 = -1$; $\phi_2 = 0$; $\phi_3 = \hat{x}$; $\phi_4 = \hat{x} \hat{y}$; $\phi_5 = - (1 + \hat{x}^2)$; $\phi_6 = \hat{y}$ and $\eta_1 = 0$; $\eta_2 = -1$; $\eta_3 = \hat{y}$; $\eta_4 = 1 + \hat{y}^2$; $\eta_5 = -\hat{x} \hat{y}$; and $\eta_6 = -\hat{x}$. Applying similar constraints on optical flow of rigid motion was first introduced by G. Adiv in [1]. However, there the optical flow is a pre-processing step to be followed by structure and motion estimation, while our formulation uses the rigid motion model in the optimality criterion of the optical flow estimation process. Since the optical flow induced by camera rotation is independent of the depth it has a more global nature and therefore one may wish to penalize more severely for changes in rotation when considering the smoothness term. This can be done by scaling the basis functions multiplying the rotation coefficients by a factor between 0 and 1. Such a factor would require larger changes in the coefficients in order to achieve the same overall influence of
the rotation on the optical flow. Such larger changes in the corresponding coefficients would be suppressed by the regularization term and therefore achieve a more global effect of the rotation. Note, that assuming rigid motion one could also extract the depth profile (up to scaling) from the above coefficients.

4.2.4 Pure translation motion model

A special case of the rigid motion scenario can be thought of when we limit the motion to simple translation. In this case we have,

\[
\begin{align*}
u &= -\theta_1 + \theta_3 \hat{x} \\
v &= -\theta_2 + \theta_3 \hat{y}.
\end{align*}
\] (4.12)

The Euler-Lagrange equations of the rigid motion still applies in this case when considering only the first \( n = 3 \) coefficients and corresponding basis functions.

4.2.5 Constant motion model

The constant motion model includes only \( n = 2 \) coefficients, with

\[
\begin{align*}
\phi_1 &= 1; \quad \phi_2 = 0 \\
\eta_1 &= 0; \quad \eta_2 = 1,
\end{align*}
\] (4.13)

as basis functions. For this model there are two coefficients to solve for \( A_1 \) and \( A_2 \) which are the optic flow components \( u \) and \( v \), respectively. In this case we obtain the familiar variational formulation where we solve for the \( u \) and \( v \) components. This fact can also be seen by substitution of Equation (4.13) into Equation (4.7), that yields the Euler-Lagrange equations used, for example, in [10] (without the gradient constancy term).
4.3 Numerical scheme

In our numerical scheme we use a multi-resolution solver, by down sampling the image data with a standard factor of 0.5 along the $x$ and $y$ axes between the different resolutions. The solution is interpolated from coarse to fine resolution. Similar techniques to overcome the intrinsic non-convex nature of the resulting optimization problem were used, for example, in [10] \(^1\). At the lowest resolution, we start with the initial guess of $A_i = 0$, $i = 1, ..., n$. From this guess, the solution is iteratively refined and the coefficients are interpolated to become the initial guess at the next higher resolution and the process is repeated until the solution at the finest resolution is reached. At each resolution, three loops of iterations are applied. At the outer loop with iteration variable $k$, we freeze the brightness constancy linear approximation in the Euler-Lagrange equations

$$
\Psi' \left( (I_z^{k+1})^2 \right) \cdot I_z^{k+1} \cdot \left( (I_x^+)^k \phi_q + (I_y^+)^k \eta_q \right) - \
\alpha \cdot \text{div} \left( \Psi' \left( \sum_{i=1}^{n} \|\nabla A^{k+1}_i\|^2 \right) \nabla A^{k+1}_q \right) = 0. \quad (4.14)
$$

Inserting the first terms of the Taylor expansion

$$
I_z^{k+1} \approx I_z^k + (I_x^+)^k \sum_{i=1}^{n} dA^k_i \phi_i + (I_y^+)^k \sum_{i=1}^{n} dA^k_i \eta_i, \quad (4.15)
$$

where $A_i^{k+1} = A_i^k + dA^k_i$. The second inner loop – fixed point iteration – deals with the nonlinearity of $\Psi$ with iteration variable $l$, it uses the expressions of $\Psi'$ from the previous iteration in both the data and smoothness terms, while the rest of the equation is written with respect to the $l + 1$ iteration

$$
(\Psi')^{kl}_{Data} d_q \left( I_z^k + \sum_{i=1}^{n} dA^{k,l+1}_i d_i \right) - 
$$

\(^1\) We compare our model and results to [10] since, to the best of our knowledge, that paper reports the most accurate flow field results for the Yosemite without clouds sequence.
\[
\alpha \cdot \text{div} \left( (\Psi')^{k,l}_{\text{Smooth}} \hat{\nabla} \left( A^k_q + dA^{k,l+1}_q \right) \right) = 0, \quad (4.16)
\]

where,
\[
d_m := (I^+_x)^k \phi_m + (I^+_y)^k \eta_m, \quad (4.17)
\]

\[
(\Psi')^{k,l}_{\text{Data}} := \Psi' \left( (I^+_z)^k + \sum_{i=1}^n dA^{k,l}_i d_i \right)^2, \quad (4.18)
\]

and
\[
(\Psi')^{k,l}_{\text{Smooth}} := \Psi' \left( \sum_{i=1}^n \left\| \hat{\nabla} \left( A^k_i + dA^{k,l}_i \right) \right\|^2 \right). \quad (4.19)
\]

At this point, we have for each pixel \( n \) linear equations with \( n \) unknowns, the increments of the coefficients of the model parameters. The linear system of equations is solved on the sequence volume using Gauss-Seidel iterations. Each Gauss-Seidel iteration involves the solution of \( n \) linear equations for each pixel as described by Equation (4.16). The discretization uses two point central difference for the flow components and four point central difference for the image derivatives as suggested in [4].

4.4 Experimental Results

In this section we compare our optical flow computation results to the best published results. For test sequences with ground truth, we use the standard measures of Average Angular Error (AAE) and Standard Deviation (STD). Our results are measured over all the pixels (100% dense). The angle is defined as:
\[
\theta = \cos^{-1} \left( \frac{uu_y + vv_y + 1}{\sqrt{(u^2 + v^2 + 1)(u^2 + v^2)}} \right), \quad (4.20)
\]
where, $u$ and $v$ are the estimated optical flow components, and $u_g$ and $v_g$ represent the ground truth optical flow. The AAE is the average and STD is the standard deviation of $\theta$ over the image domain.

4.4.1 Parameter settings

The parameters were set experimentally by an optimization process which numerically minimizes the weighted sum of AAE and STD measured on the Yosemite sequence. Such a parameter optimization or training process is usual in many other papers, for example [46]. References [10, 45] also seem to have an optimal parameter setting since in their parameter sensitivity analysis, each parameter change in any direction results in an increase of the AAE measure for the Yosemite sequence. We have found slightly different parameter settings for the 2D and 3D smoothness cases as shown in Table 4.1, where 2D refers to smoothness term with only spatial derivatives, while 3D refers to spatio-temporal smoothness term that couples the solution of the optic flow field at different frames (also known as the "two frames" versus "multi-frame" formulation). Here $\sigma$ denotes the standard deviation of the 2D Gaussian pre-filter used for pre-processing the image sequence. We used 60 iterations of the outer loop in the 3D method and 80

<table>
<thead>
<tr>
<th>Method</th>
<th>$\alpha$</th>
<th>$\rho$</th>
<th>$\omega^2_t$</th>
<th>$\varepsilon$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant motion (3D)</td>
<td>16.0</td>
<td>-</td>
<td>9.0</td>
<td>0.001</td>
<td>0.8</td>
</tr>
<tr>
<td>Affine (2D)</td>
<td>58.3</td>
<td>0.858</td>
<td>0</td>
<td>0.001</td>
<td>0.8</td>
</tr>
<tr>
<td>Pure translation (2D)</td>
<td>51.0</td>
<td>0.575</td>
<td>0</td>
<td>0.001</td>
<td>0.8</td>
</tr>
<tr>
<td>Affine (3D)</td>
<td>32.9</td>
<td>1.44</td>
<td>0.474</td>
<td>0.001</td>
<td>0.8</td>
</tr>
<tr>
<td>Rigid motion (3D)</td>
<td>54.6</td>
<td>1.42</td>
<td>0.429</td>
<td>0.001</td>
<td>0.8</td>
</tr>
<tr>
<td>Pure translation (3D)</td>
<td>23.6</td>
<td>1.22</td>
<td>0.688</td>
<td>0.001</td>
<td>0.8</td>
</tr>
</tbody>
</table>
iterations in the 2D method, 5 inner loop iterations and 10 Gauss-Seidel iterations.

4.4.2 Yosemite sequence

We applied our method to the Yosemite sequence without clouds (available at [5]), with four resolution levels. Table 4.2 shows our results relative to the best published ones. As seen in the table, our method achieves a better reconstructed solution compared to all other reported results for this sequence, both for the 2D and 3D cases. In fact, our result for the 2D case is good even compared to 3D results from the literature. Figures 4.4 and 4.5 show the solution of the affine parameters from which one can observe the trends of the flow changes with respect to the $x$ and $y$ axes. The solution of the pure translation parameters is shown in figure 4.6. The depth discontinuities in the scene are sharp and evident in all the parameters. Figure 4.1 shows the ground truth optical flow and the results with the pure translation model. Figure 4.2 shows the image of the angular errors. Figure 4.3 shows both the histogram and the cumulative probability of the angular errors. Both figures demonstrating the typically lower angular errors of our method. Table 4.3 summarizes the noise sensitivity results of our method. We also coupled the affine over-parameterized model with our previous work on joint optic flow computation and denoising presented in [44] by iterating between optical flow computation on the denoised sequence and denoising with the current optical flow. This coupling provides a model with robust behavior under noise, that obtains better AAE measure under all noise levels compared to the best published results.
Table 4.2: Yosemite sequence without clouds.

<table>
<thead>
<tr>
<th>Method</th>
<th>AAE</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Papenberg et al. 2D smoothness [45]</td>
<td>1.64°</td>
<td>1.43°</td>
</tr>
<tr>
<td>Brox et al. 2D smoothness [10]</td>
<td>1.59°</td>
<td>1.39°</td>
</tr>
<tr>
<td>Memin-Perez [35]</td>
<td>1.58°</td>
<td>1.21°</td>
</tr>
<tr>
<td>Roth et al. [46]</td>
<td>1.47°</td>
<td>1.54°</td>
</tr>
<tr>
<td>Bruhn et al. [12]</td>
<td>1.46°</td>
<td>1.50°</td>
</tr>
<tr>
<td>Amiaz et al. 2D smoothness (Over-fine x4) [51]</td>
<td>1.44°</td>
<td>1.55°</td>
</tr>
<tr>
<td>Farneback [21]</td>
<td>1.40°</td>
<td>2.57°</td>
</tr>
<tr>
<td>Liu et al. [33]</td>
<td>1.39°</td>
<td>2.83°</td>
</tr>
<tr>
<td><strong>Our method affine 2D smoothness</strong></td>
<td>1.18°</td>
<td>1.31°</td>
</tr>
<tr>
<td>Govindu [23]</td>
<td>1.16°</td>
<td>1.17°</td>
</tr>
<tr>
<td>Farneback [22]</td>
<td>1.14°</td>
<td>2.14°</td>
</tr>
<tr>
<td><strong>Our method constant motion 3D smoothness</strong></td>
<td>1.07°</td>
<td>1.21°</td>
</tr>
<tr>
<td>Papenberg et al. 3D smoothness [45]</td>
<td>0.99°</td>
<td>1.17°</td>
</tr>
<tr>
<td>Brox et al. 3D smoothness [10]</td>
<td>0.98°</td>
<td>1.17°</td>
</tr>
<tr>
<td><strong>Our method rigid motion 3D smoothness</strong></td>
<td>0.96°</td>
<td>1.25°</td>
</tr>
<tr>
<td><strong>Our method affine 3D smoothness</strong></td>
<td>0.91°</td>
<td>1.18°</td>
</tr>
<tr>
<td><strong>Our method pure translation 3D smoothness</strong></td>
<td>0.85°</td>
<td>1.18°</td>
</tr>
</tbody>
</table>

Table 4.3: Yosemite without clouds - noise sensitivity results.

<table>
<thead>
<tr>
<th>( \sigma_n )</th>
<th>Our method (affine)</th>
<th>Our method (affine) coupled with [44]</th>
<th>Results reported in [10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.91 ± 1.18°</td>
<td>0.93 ± 1.20°</td>
<td>0.98 ± 1.17°</td>
</tr>
<tr>
<td>20</td>
<td>1.59 ± 1.67°</td>
<td>1.52 ± 1.48°</td>
<td>1.63 ± 1.39°</td>
</tr>
<tr>
<td>40</td>
<td>2.45 ± 2.29°</td>
<td>2.02 ± 1.76°</td>
<td>2.40 ± 1.71°</td>
</tr>
</tbody>
</table>
Figure 4.1: Optic flow of the Yosemite sequence. Left - ground truth. Right - Our method with the pure translation model.

Figure 4.2: Images of the angular error (sky region excluded), white indicates zero error and black indicates an error of 3 degrees or above. Left - our method with the affine model. Middle - our method with the pure translation model. Right - the method of [10], optic flow field courtesy of the authors.
Figure 4.3: Histogram (left) and Cumulative probability (right) of the angular errors. Solid - pure translation model. Dash dot - Affine model. Dotted - Optic flow obtained in [10] courtesy of the authors.

Figure 4.4: Solution of the affine parameters for the Yosemite sequence - from left to right $A_1$, $A_2$, $A_3$. 
Figure 4.5: Solution of the affine parameters for the Yosemite sequence - from left to right $A_4, A_5, A_6$.

Figure 4.6: Solution of the pure translation parameters for the Yosemite sequence - from left to right $A_1, A_2, A_3$. 

4.4.3 Synthetic piecewise constant affine flow example

For illustration purposes we also considered the piecewise affine flow over an image of size 100 × 100 having the ground truth shown in Figure 4.8, and given by

For $x < 40$,

$u = -0.8 - 1.6(x - 50)/50 + 0.8(y - 50)/50$,

$v = 1.0 + 0.65(x - 50)/50 - 0.35(y - 50)/50$.

For $x \geq 40$,

$u = 0.48 - 0.36(x - 50)/50 - 0.6(y - 50)/50$,

$v = 0.3 - 0.75(x - 50)/50 - 0.75(y - 50)/50$.

The two images used for this test were obtained by sampling a grid of size 100×100 from frame 8 of the Yosemite sequence (denoted $I_{\text{yosemite}}$). The second image $I_2(x, y) = I_{\text{yosemite}}(x + \Delta x, y + \Delta y)$ and the first image is sampled at warped locations $I_1(x, y) = I_{\text{yosemite}}(x + \Delta x + u, y + \Delta y + v)$ using bilinear interpolation. The constant shift values are: $\Delta x = 79$ and $\Delta y = 69$. The two images obtained are displayed in Figure 4.7. The results exhibited in Table 4.4 show that our method with the affine over-parametrization outperforms the method of [10]. This is to be expected since the true flow is not piecewise constant and the smoothness term in [10] penalizes for changes from the constant flow model, whereas, the affine over-parametrization model solves the optimization problem in the (correct) affine space in which it accurately finds the piecewise constant affine parameters solution of the problem, as shown in figures 4.9 and 4.10. One can notice that
the discontinuity at pixel $x = 40$ is very well preserved due to the effective edge preserving $L_1$ based optimization. The resulting optical flow shown in Figure 4.11 accurately matches the ground truth.

![Figure 4.7: The two images: Left - first frame. Right - second frame.](image)

4.4.4 Flower garden sequence

We also applied our method to the real image sequence "flower garden". The results obtained by the 2D affine model are shown in Figure 4.12. We have checked our results by manually marking the coordinates of several corresponding points on the two images and comparing the motion with the optical flow results. We have found good match between manual tracking and the computed flow. The tree moves with a velocity of about $-5.7\text{pixels/frame}$ along the $x$ direction. The whole scene has low velocity in the $y$ direction (the computed $v$ is between $-0.9$ and $1.4\text{pixel/frame}$). The $y$ component of the flow is about $1\text{pixel/frame}$ in the upper section of the tree and almost zero in the lower section. In the area of the garden, the velocity is decreasing as the distance from the camera increases from about $-3\text{pixels/frame}$ in the right lower section ($-2$ on the left)
Figure 4.8: Synthetic piecewise affine flow - Ground truth. Left - $u$ component. Right - $v$ component.

Table 4.4: Synthetic piecewise affine flow test.

<table>
<thead>
<tr>
<th>Method</th>
<th>AAE</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our implementation of [10] excluding the gradient constancy term</td>
<td>1.48°</td>
<td>2.28</td>
</tr>
<tr>
<td>Our method with the affine flow model</td>
<td>0.88°</td>
<td>1.67</td>
</tr>
</tbody>
</table>

Figure 4.9: Solution of the affine parameters - from left to right $A_1, A_2, A_3$.

Figure 4.10: Solution of the affine parameters - from left to right $A_4, A_5, A_6$. 

Technion - Computer Science Department - Ph.D. Thesis PHD-2007-08 - 2007
Figure 4.11: Ground truth for the synthetic test (left) and computed flow by the affine over-parametrization model (right).
to about $-1 \text{pixel/frame}$ in the upper part and in the area of the houses. The computed flow by the pure translation over-parameterized model shown in Figure 4.13 produces similar flow field as the results with the affine model. The solution of the three pure translation model parameters is shown in Figure 4.14.

![Figure 4.12: Flower garden sequence: First image (left). Optical flow computed by the 2D affine over-parametrization model, $u$ (middle) and $v$ (right).](image)

4.4.5 Road sequence

The two frames of this real sequence courtesy of the authors of [54] are shown in Figure 4.15. The computed flow by our 2D over-parameterized affine model is shown in Figure 4.16. The motion compensated image difference is shown in Figure 4.17, our motion compensated reconstruction ratio is $PSNR = 38.17dB$ (more than 1.13$dB$ better than the results reported in to [54] with for the affine model and 0.58$dB$ better than their quadratic model). The measure of the motion compensated difference might be misleading for comparing the quality of different optical flow algorithms since one can generate a globally optimal algorithm for this measure which produces meaningless optical flow results. For example: suppose we find for each pixel in one image a corresponding pixel with the same gray value in the other image. In this case, the resulting optical flow is perfect in terms of $PSNR$, however, it can be arbitrary vectors connecting pixels with the same gray
Figure 4.13: Flower garden sequence: Optical flow computed by the 2D pure translation over-parametrization model, $u$ (left) and $v$ (right).

Figure 4.14: Flower garden sequence: The solution of the pure translation parameter. From left to right $A_1, A_2, A_3$. 
level belonging to different objects in the scene. For this example there is no
ground truth motion and therefore we use this measure in order to compare with
the results reported in [54].

![Road sequence. Frame 1 - Left. Frame 2 - Right.](image)

**Figure 4.15: Road sequence. Frame 1 - Left. Frame 2 - Right.**

### 4.5 Summary

We here introduce a novel over-parameterized variational framework for ac-
curately solving the optical flow problem. The flow field is represented by a
general space-time model. The proposed approach is useful and highly flexible
in the fact that each pixel has the freedom to choose its own set of model pa-
rameters. Subsequently, the decision on the discontinuity locations of the model
parameters, is resolved within the variational framework for each sequence. Many
variational approaches can be regarded as special cases of our method when one
selects a constant motion model. Observe however that in most scenarios, the
Figure 4.16: Optical flow of the road sequence with the affine overparameterization model. $u$ is shown on the left and $v$ on the right.

Figure 4.17: Difference of the frames. Left - Difference of the original frames ($PSNR = 23.73dB$). Right - Motion compensated difference by the estimated optical flow ($PSNR = 38.17dB$). Both images are scaled by the same linear transformation from errors to graylevels.
optical flow would be better represented by a piecewise constant affine model or a rigid motion model rather than a piecewise constant flow. Therefore, compared to existing variational techniques, the smoothness penalty term modeled by the proposed over-parametrization models yields better optic flow recovery performance as demonstrated by our experiments. In this work, the derivation is for general space-time basis functions. We limited our experiments to spatial basis functions. Future work will carry at experiments with spatio-temporal basis functions. Incorporating learning of the basis functions (dictionaries) for specific scenes could be of great interest and useful for video compression. As a consequence of this work, motion segmentation based on optical flow should generally be replaced by segmentation in the higher dimensional parameter space as suggested by our initial results presented herein for the synthetic sequence. Although the models suggested in this paperwork were all over-parameterized, an under-parameterized model might also be used in this framework, for example in case one has prior knowledge regarding constraints between the $u$ and $v$ components (as in stereo matching or when we know that the optic flows is radial).
Chapter 5

Causal Camera Motion Estimation by Condensation and Robust Statistics Distance Measures

5.1 Introduction

While the vision community struggled with the difficult problem of estimating motion and structure from a single camera generally moving in 3D space (see [13]), the robotics community independently addressed a similar estimation problem known as Simultaneous Localization and Mapping (SLAM) using odometry, laser range finders, sonars and other types of sensors together with further assumptions such as planar robot motion. Recently, the vision community has adopted the SLAM name and some of the methodologies and strategies from the robotics community. Vision based SLAM has been proposed in conjunction with an active stereo head and odometry sensing in [17], where the stereo head actively searched for old and new features with the aim of improving the SLAM accuracy. In [16] the more difficult issue of localization and mapping based on data from a single passive camera is treated. The camera is assumed to be calibrated and some features with known 3D locations are assumed present and these features impose
a metric scale on the scene, enable the proper use of a motion model, increase the estimation accuracy and avoid drift. These works on vision based SLAM employ an Extended Kalman Filter (EKF) approach where camera motion parameters are packed together with 3D feature locations to form a large and tightly coupled estimation problem. The main disadvantage of this approach is that even a single outlier in measurement data can lead to a collapse of the whole estimation problem. Although there are means for excluding problematic feature points in tracking algorithms, it is impossible to completely avoid outliers in uncontrolled environments. These outliers may result from mismatches of some feature points which are highly likely to occur in cluttered environments, at depth discontinuities or when repetitive textures are present in the scene. Outliers may exist even if the matching algorithm performs perfectly when some objects in the scene are moving. In this case multiple-hypothesis estimation as naturally provided by particle filters is appropriate. The estimation of the stationary scene structure together with the camera ego-motion is the desired output under the assumption that most of the camera’s field of view looks at a static scene. The use of particle filters in SLAM is not new. Algorithms for FastSLAM [37] employed a particle filter for the motion estimation, but their motivation was mainly computational speed and robust estimation methodology was neither incorporated nor tested. In [36] a version of FastSLAM addressing the problem of data association between landmarks and measurements is presented. However, the solution to the data association problem provided there does not offer a solution to the problem of outliers since all landmarks are assumed stationary and every measurement is assumed to correctly belong to one of the real physical landmarks. Other works like e.g. [16] employed condensation only in initialization of distances of new feature points
before their insertion into the EKF. However the robustness issue is not solved in this approach since the motion and mapping are still provided by the EKF. In [52] the pose of the robot was estimated by a condensation approach. However, here too the algorithm lacked robust statistics measures to effectively reject outliers in the data. Furthermore the measurements in this work were assumed to be provided by laser range finders and odometric sensors. In this chapter we propose a new and robust solution to the basic problem of camera motion estimation from known 3D feature locations, which has practical importance of its own. The full SLAM problem is then addressed in the context of supplementing this basic robust camera motion estimation approach for simultaneously providing additional 3D scene information.

5.2 Problem Formulation

Throughout this chapter it is assumed that the camera is calibrated. This assumption is commonly made in previous works on vision based SLAM. A 3D point indexed by i in the camera axes coordinates, \((X_i(t), Y_i(t), Z_i(t))^T\) projects to the image point \((x_i(t), y_i(t))^T\) at frame time \(t\) via some general projection function denoted \(\Pi\)

\[
\begin{pmatrix}
x(t) \\
y(t)
\end{pmatrix} = \Pi \begin{pmatrix} X(t) \\
Y(t) \\
Z(t)
\end{pmatrix}.
\] (5.1)
The camera motion between two consecutive frames is represented by a rotation matrix $R(t)$ and a translation vector $V(t)$. Hence for a static point in the scene

$$\begin{pmatrix} X(t) \\ Y(t) \\ Z(t) \end{pmatrix} = R(t) \begin{pmatrix} X(t-1) \\ Y(t-1) \\ Z(t-1) \end{pmatrix} + V(t). \quad (5.2)$$

The rotation is represented using the exponential canonical form $R(t) = e^{\hat{\omega}(t)}$ where $\omega(t)$ represents the angular velocity between frames $t-1$ and $t$, and the exponent denotes the matrix exponential. The hat notation for some 3D vector $q$ is defined by

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} ; \hat{q} = \begin{pmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{pmatrix} \quad (5.3)$$

The matrix exponential of such skew-symmetric matrices may be computed using the Rodrigues’ formula

$$e^{\hat{\omega}} = I + \frac{\hat{\omega}}{\|\omega\|} \sin (\|\omega\|) + \frac{\hat{\omega}^2}{\|\omega\|^2} (1 - \cos (\|\omega\|)) \quad (5.4)$$

Let us denote by $\Omega(t)$ and $T(t)$ the overall rotation and translation from some fixed world coordinate system to the camera axes

$$\begin{pmatrix} X_i(t) \\ Y_i(t) \\ Z_i(t) \end{pmatrix} = e^{\hat{\Omega}(t)} \begin{pmatrix} X_{iWorld} \\ Y_{iWorld} \\ Z_{iWorld} \end{pmatrix} + T(t). \quad (5.5)$$

Equation (5.5) describes the pose of the world relative to the camera. The camera pose relative to the world is given by $\Omega_{\text{Camera}}^t = -\Omega(t)$ and $T_{\text{Camera}}^t = -e^{-\hat{\Omega}(t)}T(t)$. Using equations (5.2), (5.5) and (5.5) written one sample backward

$$\Omega(t) = \log \left(e^{-\hat{\Omega}(t)}e^{\hat{\Omega}(t-1)}\right) \quad (5.6)$$

$$T(t) = e^{-\hat{\Omega}(t)}T(t-1) + V(t)$$
where, \( q = \text{log}(A) \) denotes the inverse of the matrix exponential of the skew symmetric matrix \( A \) such that \( A = e^q \) (i.e. inverting the Rodrigues’ formula). Let us define the robust motion from structure estimation problem: given matches of 2D image feature points to known 3D locations, estimate the camera motion in a robust framework accounting for the possible presence of outliers in measurement data.

### 5.2.1 Dynamical Motion Model

One can address the camera motion estimation problem with no assumptions on the dynamical behavior of the camera (motion model), thus using only the available geometric information in order to constrain the camera motion. This is equivalent to assuming independent and arbitrary viewpoints at every frame. In most practical applications though, physical constraints result in high correlation of pose between adjacent frames. For example, a camera mounted on a robot traveling in a room produces smooth motion trajectories unless the robot hits some obstacle or collapses. The use of a proper motion model accounts for uncertainties, improves the estimation accuracy, attenuates the influence of measurement noise and helps overcome ambiguities (which may occur if at some time instances, the measurements are not sufficient to uniquely constrain camera pose, see [5] and [6]). Throughout this chapter, the motion model assumes constant velocity with acceleration disturbances, as follows

\[
\omega(t) = \omega(t - 1) + \dot{\omega}(t)
\]

\[
V(t) = V(t - 1) + \dot{V}(t)
\]

If no forces act on the camera the angular and translation velocities are constant. Accelerations result from forces and moments which are applied on the camera,
and these being unknown are treated as disturbances (recall that the vectors $\omega(t), V(t)$ are velocity terms and the time is the image frame index). Acceleration disturbances are modeled here probabilistically by independent white Gaussian noises

$$\dot{\omega}(t) \sim N(0, \sigma_\omega)$$
$$\dot{V}(t) \sim N(0, \sigma_V)$$

where, $\sigma_\omega$ and $\sigma_V$ denote expected standard deviations of the angular and translational acceleration disturbances.

### 5.3 Robust Motion from Structure by Condensation

In this section we present the proposed condensation based algorithm designed for robust camera 3D motion estimation. A detailed description of condensation in general and its application to contour tracking can be found in [29]. The state vector of the estimator at time $t$, denoted by $s_t$, includes all the motion parameters $s_t = \left( \Omega(t) \ T(t) \ \omega(t) \ V(t) \right)^T$. The state vector is of length 12. The state dynamics are generally specified in the condensation framework by the probability distribution function $p(s_t \mid s_{t-1})$. Our motion model is described by equations (4),(5),(6). All measurements at time $t$ are denoted compactly as $z(t)$. The camera pose is defined for each state $s_t$ separately, with the corresponding expected projections being tested on all the visible points in the current frame

$$\left( \begin{array}{c} x(t) \\ y(t) \end{array} \right) = \Pi \left( e^{\Omega(t)} \begin{pmatrix} X_i^{World} \\ Y_i^{World} \\ Z_i^{World} \end{pmatrix} + T(t) \right)$$
The influence of the measurements is quantified by \( p(z(t) \mid s_t) \). This is the conditional Probability Distribution Function (PDF) of measuring the identified features \( z(t) \) when the true parameters of motion correspond to the state \( s_t \). The conditional PDF is calculated as a function of the geometric error, which is the distance denoted by \( d_i \) between the projected 3D feature point location on the image plane and the measured image point. If the image measurement errors are statistically independent random variables with zero mean Gaussian PDF, then up to a normalizing constant

\[
p(z \mid s) = d \exp \left( -\sum_{i=1}^{N_{\text{points}}} \frac{d_i^2}{2\sigma^2 N_{\text{points}}} \right)
\]

Where \( N_{\text{points}} \) is the number of visible feature points and \( \sigma \) is the standard deviation of the measurement error (about 1 pixel). Since outliers have large \( d_i \) values even for the correct motion, the quadratic distance function may be replaced by a robust distance function \( \rho (d_i^2) \) see e.g. [26]

\[
p(z \mid s) = d \exp \left( -\sum_{i=1}^{N_{\text{points}}} \frac{\rho (d_i^2)}{2\sigma^2 N_{\text{points}}} \right)
\] (5.9)

\[
\rho (d^2) = \frac{d^2}{1 + d^2/L^2}.
\] (5.10)

If some feature point is behind the camera (this occurs when its 3D coordinates expressed in camera axes have a negative Z value), clearly this feature should not have been visible and hence its contribution to the sum is set to the value

\[
\lim_{d_i \to \infty} \rho (d_i^2) = L^2
\]

. The influence of every feature point on the PDF is now limited by the parameter \( L \). The choice of \( L \) reflects a threshold value between inliers and outliers. In order
to understand why robustness is achieved using such distance functions, let us consider the simpler robust distance function, the truncated quadratic

\[
\rho(d^2) = \begin{cases} 
    d^2 & \text{if } d^2 < A^2 \\
    A^2 & \text{Otherwise}
\end{cases}
\]

where, \(A\) is the threshold value between inliers and outliers. Using this \(\rho\) function in equation (5.9) yields

\[
p(z \mid s) = \exp\left(-\frac{\sum_{i \in \text{Inlier Points}} d_i^2 + \sum_{i \in \text{Outlier Points}} A^2}{2\sigma^2 N_{\text{points}}}ight) = \exp\left(-\frac{\sum_{i \in \text{Inlier Points}} d_i^2 + A^2(\text{NumberOfOutliers})}{2\sigma^2 N_{\text{points}}}ight)
\]

Maximizing this PDF (a maximum likelihood estimate) is equivalent to minimizing the sum of the two terms, the first is the sum of the quadratic distances at the inlier points and the second term is proportional to the number of outliers. The robust distance function of equation (5.10) is similar to the truncated quadratic, with a smoother transition between the inliers and outliers (see [6] and [7] for an analysis of functions used in robust statistics and their use for image reconstruction and for the calculation of piecewise-smooth optical flow fields). Let us summarize the proposed algorithm for robust 3D motion estimation from known structure:

Initialization- Sample \(N\) states \(S_0^{(n)}, n = 1...N\) from the prior PDF of \(\omega(0), V(0)\) and \(\Omega(0), T(0)\). Initialize \(\pi_0^{(n)}\) with the PDF corresponding to each state. At every time step \(t=1,2,\ldots\):

- Sample \(N\) states \(\tilde{S}_{t-1}^{(n)}\) copied from the states \(S_{t-1}^{(n)}\) with probabilities \(\pi_{t-1}^{(n)}\).

- Propagate the sampled states using equations (5.8),(5.7),(5.6) to obtain \(S_t^{(n)}\).
• Incorporate the measurements to obtain $\pi_t^{(n)} = p \left(z_t \mid s_t^{(n)}\right)$ using equations (5.9),(5.10). Then normalize by the appropriate factor so that
$$\sum_{n=1}^{N} \pi_t^{(n)} = 1$$

Extract the dominant camera motion from the state $s_t^{(n)}$ corresponding to the maximum of $\pi_t^{(n)}$: $\Omega_{(t)}^{Camera} = -\Omega_{(t)}^{(n)}$; $T_{(t)}^{Camera} = -e^{-\Omega_{(t)}^{(n)}} T_{(t)}^{(n)}$

Code written in C++ implementing the algorithm of this section can be found in [25]. It can run in real time on a Pentium 4, 2.5GHz processor, with 30Hz sampling rate, 1000 particles and up to 200 instantaneously visible feature points.

5.4 Application to SLAM

This section describes various possible solutions to the robust SLAM problem.

5.4.1 SLAM in a Full Condensation Framework

The most comprehensive solution to robust SLAM is the packing of all the estimated parameters into one large state and solve using a robust condensation framework. The state is composed of the motion parameters and each feature contributes three additional parameters for its 3D location. As stated in [17], this solution is very expensive computationally due to the large number of particles required to properly sample from the resulting high dimensional space.

5.4.2 SLAM in a Partially Decoupled Scheme

The research on vision based SLAM tends to incorporate features with known 3D locations in the scene. The simplest method for incorporating the
proposed robust motion from structure algorithm into SLAM is in a partially
decoupled block scheme in which the features with known 3D locations are the
input to the robust motion estimation block of section 3. Structure of other fea-
tures in the scene can be recovered using the estimated motion and the image
measurements. Assuming known 3D motion, the structure of each feature can be
estimated using an EKF independently for each feature (similar to FastSlam in
[37]). If enough features with known structure are available in the camera field
of view at all times (few can be enough as shown in the experiments section),
then this method can work properly. It may be practical for robots moving in
rooms and buildings to locate known and uniquely identifiable features (fiducials)
at known locations. When the motion estimation is robust, the independence of
the estimators for the structure of the different features guarantees the robustness
of the structure recovery as well.

![Diagram](Diagram.png)

Figure 5.1: Scheme for partially decoupled SLAM.

### 5.4.3 SLAM with Robust Motion Estimation and Triangulation

In this section we propose a solution to the robust SLAM problem in a con-
densation framework with a state containing motion parameters only. In the mea-
surement phase, features with known locations have their 3D structure projected
on the image plane, features with unknown structure have their 3D structure re-
constructed using triangulation (see [24] chapter 11) and the geometric error is
measured by projecting this structure back on the image plane. The information regarding the camera pose in the current and previous frames is embedded in each state hypothesis of the condensation algorithm which together with the corresponding image measurements form the required information for the triangulation process. Triangulation can be performed from three views, where the third view is the first appearance of the feature.

5.5 Experimental Results

In this section, we demonstrate the robustness of our method compared to the EKF approach on both synthetic and real image sequences.

5.5.1 Synthetic tests

It has been experimentally found using synthetic tests that robustness with the proposed method is maintained with up to about 33 percent of outliers. The proposed algorithm is compared with the results of the EKF approach in [13] which is run with the code supplied in [30]. The robust motion estimation used triangulation in three frames as described in section 4.3. The 3D structure was unknown to both algorithms. The outlier points are randomly chosen and remain fixed throughout the sequence, these points are given random image coordinates uniformly distributed in the image range (see examples in [25]). The rotation errors are compactly characterized by

$$
\| I - (e^{\hat{\Omega}_{\text{True}}}^T e^{\hat{\Omega}_{\text{Estimated}}}^T)^T \|_{Frobenius}^2
$$

The estimation results are shown in figures 5.2, 5.3, 5.4. With 33 percent of outliers, the EKF errors are unacceptable while the proposed method maintains reasonable accuracy.
Figure 5.2: The translation error with no outliers in the data.
Figure 5.3: The rotation error with no outliers in the data.
Figure 5.4: The rotation errors with 33 percent of outliers.
5.5.2 Real Sequence Example

In this section a test consisting of 340 frames is described in detail. More sequences can be found in [40]. Small features (fiducials) were placed in known 3D locations (see Table 1) on the floor and on the walls of a room (see Fig. 2). Distances were measured with a tape having a resolution of 1 millimeter (0.1 cm). The fixed world coordinate system was chosen with its origin coinciding with a known junction on the floor tiles, the $X$ and $Z$ axes on the floor plane and parallel to the floor tiles and the $Y$ axis pointing downwards (with $-Y$ measuring the height above the floor).

Table 5.1: Scene fiducial geometry.

<table>
<thead>
<tr>
<th>Serial number</th>
<th>Type</th>
<th>Color</th>
<th>$X_{World}$</th>
<th>$Y_{World}$</th>
<th>$Z_{World}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ball</td>
<td>Blue</td>
<td>30</td>
<td>-0.7</td>
<td>180</td>
</tr>
<tr>
<td>2</td>
<td>Ball</td>
<td>Green</td>
<td>30</td>
<td>-0.7</td>
<td>210</td>
</tr>
<tr>
<td>3</td>
<td>Ball</td>
<td>Yellow</td>
<td>-60</td>
<td>-0.7</td>
<td>240</td>
</tr>
<tr>
<td>4</td>
<td>Ball</td>
<td>Light blue</td>
<td>30</td>
<td>-0.7</td>
<td>240</td>
</tr>
<tr>
<td>5</td>
<td>Ball</td>
<td>Black</td>
<td>0</td>
<td>-0.7</td>
<td>270</td>
</tr>
<tr>
<td>6</td>
<td>Ball</td>
<td>Red</td>
<td>-30</td>
<td>-0.7</td>
<td>330</td>
</tr>
<tr>
<td>7</td>
<td>Ball</td>
<td>Orange</td>
<td>60</td>
<td>-0.7</td>
<td>360</td>
</tr>
<tr>
<td>8</td>
<td>Circle</td>
<td>Light blue</td>
<td>-31</td>
<td>-100.3</td>
<td>388</td>
</tr>
<tr>
<td>9</td>
<td>Circle</td>
<td>Light blue</td>
<td>29</td>
<td>-120.7</td>
<td>492.5</td>
</tr>
</tbody>
</table>

The balls are 1.4 and the circles are 1 cm in diameter, the tiles are squares of 30x30 cm.

5.5.2.1 Camera Setup and Motion

The camera used for this test is a Panasonic NV-DS60 PAL color camera with a resolution of 720x576 pixels. The camera zoom was fixed throughout the test at the widest viewing angle. A wide field of view reduces the angular ac-
Figure 5.5: First frame.
curacy of a pixel, but enables the detection of more features (overall, [13] has experimentally found that a wide viewing angle is favorable for motion and structure estimation). The camera projection parameters at this zoom were obtained from a calibration process: 

\[ x = 938X/Z + 360.5 \]

and

\[ y = 1004Y/Z + 288.5. \]

The camera was initially placed on the floor with the optical axis pointing approximately in the Z direction of the world. The camera was moved backwards by hand on the floor plane with the final orientation approximately parallel to the initial (using the tile lines). The comparison between the robust and the EKF approach is made with both having the same motion parameters in the estimated state, the same measurements and the same knowledge of the 3D data of Table 5.1. The acceleration disturbance parameters for both methods are

\[ \sigma_\omega = 0.003 \]

\[ \sigma_V = 0.0005. \]

The number of particles is 2000 and the robust distance function parameter is \( L = 4\text{pixels} \).

5.5.2.2 Feature Tracking

The features were tracked with a Kanade-Lucas-Tomasi (KLT) type feature tracker (see [21]). The tracker was enhanced for color images by minimizing the sum of squared errors in all three RGB color channels (the standard KLT is formulated for grayscale images). The tracking windows of size 9x9 pixels were initialized in the first frame at the center of each ball and circle by manual selection. To avoid the fatal effect of interlacing, the resolution was reduced in the vertical image plane by sampling every two pixels (processing one camera field), the sub-pixel tracking results were then scaled to the full image resolution.
5.5.2.3 Feature Tracking

The results obtained by the proposed robust approach and the EKF approach are shown in figures 5.6, 5.7, 5.8, 5.9. Most of the motion is in the $Z$ direction. The final position was at approximately $Z = -60\,\text{cm}$. The robust approach estimates the value of $Z = -61\,\text{cm}$ at the end of the motion (there is some uncertainty regarding the exact location of the camera focal center), the estimated $Y$ coordinate is almost constant and equal to the camera lens center height above the floor (about $-7.4\,\text{cm}$). The trajectory estimated by the EKF is totally different with and at the end of the motion. The deviation from the expected final camera position is by two orders of magnitude higher than the expected experimental accuracy, the EKF estimation is therefore erroneous. After observing the tracking results of all the feature points, the points 1, 2, 4, 5, 6, 8 were manually selected as the inlier points (those which reasonably track the appropriate object throughout the sequence). Running again both estimators with only the inlier points, the proposed approach results are almost unchanged, while the EKF estimation changes drastically, now producing a trajectory similar to the robust approach (see figures 5.10, 5.11, 5.12, 5.13). It should be noted that the EKF estimation produces a smoother trajectory. Image plane errors between the measurements and the projected 3D structure are shown in figures 5.14, 5.15, 5.16 (corresponding to the motion estimation shown previously). The robust method exhibits low errors for most of the features and allows high errors for the outliers (this implies that algorithm can automatically separate the inliers from the outliers by checking the projection errors). The EKF approach on the other hand exhibits large errors for both inlier and outlier features. It should be noted that the outlier features are distracted from the true object due to its small size, noise,
similar objects in the background and reflections from the shiny floor. It is possible to improve the feature tracking results by using methodologies from [31], [49], [53], but good feature tracking should be complemented with a robust methodology in order to compensate for occasional mistakes. Although the deficiencies of the EKF approach are mentioned in [13], [16], [17], no examples are given and no remedies are suggested in the camera motion estimation literature. As anonymous reviewers have suggested, we examined two methods of making the EKF solution more robust: 1. By incorporating measurements only from features which have a geometric error norm below a threshold and 2. By applying the robust distance function on the norm of the geometric error of each feature. Both failed to improve the results of the EKF. Rejection of outliers in Kalman filtering may succeed if the outliers appear scarcely or when their proportion is small. In our example these conditions are clearly violated.

5.5.2.4 Structure Computation Example

Structure of unknown features in the scene can be recovered using the estimated camera motion obtained by the robust method and the image measurements in a partially decoupled scheme as explained in section 5.4. As an example, consider the middle of the letter B appearing on the air conditioner which was tracked from frame 0 to frame 50 (it is occluded shortly afterwards). The reconstructed location of this point in the world axes is: $X = 42.3cm; Y = -45.2cm; Z = 159.6cm$. The tape measure world axes location is: $X = 42.0cm; Y = -43.7cm; Z = 155cm$. The $X, Y, Z$ differences are: 0.3, 1.5 and 4.6$[cm]$ respectively. As expected, the estimation error is larger along the optical axis (approximately the world’s $Z$ axis). The accuracy is reasonable, taking into account the short
Figure 5.6: Estimated camera 3D translation using the proposed approach.
Figure 5.7: Estimated camera 3D rotation using the proposed approach.
Figure 5.8: Estimated camera 3D translation using the EKF approach.
Figure 5.9: Estimated camera 3D rotation using the EKF approach.
Figure 5.10: Estimated camera 3D translation using only the inlier points with the proposed approach.
Figure 5.11: Estimated camera 3D rotation using only the inlier points with the proposed approach.
Figure 5.12: Estimated camera 3D translation using only the inlier points with the EKF approach.
Figure 5.13: Estimated camera 3D rotation using only the inlier points with the EKF approach.
Figure 5.14: Image plane errors. Robust approach showing the 6 inliers.
Figure 5.15: Image plane errors. Robust approach showing the 3 outliers.
Figure 5.16: Image plane errors. EKF approach with all 9 features.
baseline of 19cm produced during the two seconds of tracking this feature (the overall translation from frame 0 to frame 50). As discussed in [13], a long baseline improves the structure estimation accuracy when the information is properly integrated over time.

5.6 Summary

A robust framework for camera motion estimation has been presented with extensions to the solution of the SLAM problem. The proposed algorithm can tolerate about 33 percent of outliers and it is superior in robustness relative to the commonly used EKF approach. It has been shown that a small number of visible features with known 3D structure are enough to determine the 3D pose of the camera. It may be implied from this work that some degree of decoupling between the motion estimation and structure recovery is a desirable property of SLAM algorithms which trades some accuracy loss for increased robustness. The robust distance function used in this work is symmetric for all the features with the underlying assumption that the probability of a feature to be an inlier or an outlier is independent of time. However, in most cases, a feature is expected to exhibit a more consistent behavior as an outlier or an inlier. This property may be exploited for further improvement of the algorithm’s robustness and accuracy. Also, an interesting question for future work is: How to construct fiducials which can be quickly and accurately identified in the scene for camera localization purposes.
Chapter 6

Discussion

This thesis is partially based on the papers by Tal Nir and his coauthors Alfred Bruckstein and Ron Kimmel [44, 41, 42, 43]. The goal of this work was to revisit and offer new methodologies for several motion analysis related problems in computer vision. It turned out that chapter 2 focused on modifying the data term in the functional whereas in chapter 4 we focused on improvements of the regularizer. The improvements concerning the data term originate from other sources of errors which motivated the combined optical flow and sequence restoration coupled solution scheme. In the modification of the regularizer, we introduced a model-based solution in a higher dimensional over-parameterized space. We believe that this methodology is quite general and applicable to many regularization based methods in computer vision (as shown on chapter 3) and in other areas and may introduce major changes, improvements and new results to many methodologies in the future. In fact any function regularizer may be reformulated using a model by describing the function by the model and expressing the regularization by the model coefficients, in our examples we used an $L_1$ term of the first derivatives of the coefficients, clearly, this is just an example of the use of our method. Using the combination of our modification for the data and
smoothness terms simultaneously is natural and was demonstrated to give good results in chapter 4 for the Yosemite sequence example in the noise sensitivity analysis. An interesting question regarding the over-parametrization basis functions regards the use of learned basis functions. In this context, motion patterns in space-time could be learned adaptively for a specific scene. Consider for example a camera monitoring a road junction, the learned basis function could tell us that in certain areas there is usually no motion at all (places where no vehicle is supposed to move) areas where the motion is in a certain direction, or place where the motion has typically two characteristic directions. Solving the optical flow in the learned basis function could be used to efficiently compress the video data for transmission and also for irregular event detection which may be characterized by improper representation by the learned basis functions, as could be the case when the basis functions in a certain area contain motion only in one direction, while a vehicle is now moving in an illegal direction in the same area. For segmentation purposes, the over-parameterized space may offer a richer segmentation space than the flow components, as clearly shown in the synthetic piecewise affine flow example in chapter 4. In chapter 5, we offered several suggestions for future work regarding the incorporation of full SLAM within our framework. The replacement of feature tracking data by optical flow data is also of interest. If one uses the rigid over-parameterized motion model, then the rigid motion parameters between the camera and the scene objects are obtained within the optical flow computation method. In this respect, outliers in the data represent objects in the scene which are not static. The rejection of outliers within the CONDENSATION framework with a robust statistics measurement process is therefore of high importance. The question of how to use the additional six parameters per pixels obtained by our
solution still remains open.
Bibliography


CONDENSATION

The purpose of this work is to formulate a method to determine the shape of a moving object by using a three-dimensional camera system. The method is based on a modified Kalman filter, which is used to track the movement of the object. The method is robust to noise and can be used in real-time applications.

Features (Features) are extracted from the image and used to determine the movement of the object. The method is designed to be robust to noise and can be used in real-time applications.

Outliers (Outliers) are identified and used to improve the accuracy of the tracking. The method is designed to be robust to noise and can be used in real-time applications.

Technion - Computer Science Department - Ph.D. Thesis  PHD-2007-08 - 2007
Theinkelkaf (Total variation) shows that to reduce the error of the intermediate values of the model, a more general form of the model is required when the model is applied to a new situation. The generalization of the model is achieved through the use of a new model, which is a more general form of the model used in the previous section. The new model is applied to the new situation, and the error is reduced.

In this section, we consider the generalization of the model to a new situation. The generalization of the model is achieved through the use of a new model, which is a more general form of the model used in the previous section. The new model is applied to the new situation, and the error is reduced.

In this section, we consider the generalization of the model to a new situation. The generalization of the model is achieved through the use of a new model, which is a more general form of the model used in the previous section. The new model is applied to the new situation, and the error is reduced.

In this section, we consider the generalization of the model to a new situation. The generalization of the model is achieved through the use of a new model, which is a more general form of the model used in the previous section. The new model is applied to the new situation, and the error is reduced.

In this section, we consider the generalization of the model to a new situation. The generalization of the model is achieved through the use of a new model, which is a more general form of the model used in the previous section. The new model is applied to the new situation, and the error is reduced.

In this section, we consider the generalization of the model to a new situation. The generalization of the model is achieved through the use of a new model, which is a more general form of the model used in the previous section. The new model is applied to the new situation, and the error is reduced.

In this section, we consider the generalization of the model to a new situation. The generalization of the model is achieved through the use of a new model, which is a more general form of the model used in the previous section. The new model is applied to the new situation, and the error is reduced.

In this section, we consider the generalization of the model to a new situation. The generalization of the model is achieved through the use of a new model, which is a more general form of the model used in the previous section. The new model is applied to the new situation, and the error is reduced.

In this section, we consider the generalization of the model to a new situation. The generalization of the model is achieved through the use of a new model, which is a more general form of the model used in the previous section. The new model is applied to the new situation, and the error is reduced.

In this section, we consider the generalization of the model to a new situation. The generalization of the model is achieved through the use of a new model, which is a more general form of the model used in the previous section. The new model is applied to the new situation, and the error is reduced.

In this section, we consider the generalization of the model to a new situation. The generalization of the model is achieved through the use of a new model, which is a more general form of the model used in the previous section. The new model is applied to the new situation, and the error is reduced.
The request is in Hebrew and translates to: "The thesis focuses on the development and evaluation of functional pixels for motion estimation. The functional pixel (Optical flow) method is used to estimate the motion of pixels in an image sequence. The method is based on a dense optical flow approach, which is evaluated through several experiments. The results show that the method provides accurate motion estimation, especially when applied to complex scenes. The method is shown to be efficient and robust, with potential applications in various fields such as robotics and computer vision."
5.4.1 \textit{Performing SLAM} in the full CONDENSATION model.

5.4.2 \textit{Performing SLAM} in the full CONDENSATION model partially.

5.4.3 \textit{Performing SLAM} in the full CONDENSATION model robust motion estimation and integration.

5.5 \textit{Experiments} Results.

5.5.1 Synthetic experiments.

5.5.2 Real experiments.

5.6 \textit{Summary}.

6. \textit{Discussion}.

107 \textit{Bibliography}.
1.1 MOTIVATION 3
1.2 CONTRIBUTION CENTRALITY 3
1.3 PREVIOUS WORKS 5
2.1 INTRODUCTION 9
2.2 STATEMENT 10
2.2.1 FUNCTIONAL DECLARATION OF THE FIELD OF CONSIDERATION 11
2.2.2 SERIES OF IMAGES AND RECONSTRUCTION OF THE FIELD OF CONSIDERATION 11
2.2.3 FUNCTIONAL SELECTION 13
2.3 IMPLEMENTATION 16
2.4 EXPERIMENTS RESULTS 18
2.4.1 YOSEMTI SERIES 18
2.4.2 OFFICE SERIES 22
2.5 SUMMARY 24
3.1 INTRODUCTION 25
3.2 PROBLEM DESCRIPTION 26
3.3 FUNCTIONAL ON THE FIELD OF CONSIDERATION SEPARABLE PARAMETERIZATION model 31
3.4 FUNCTIONAL ON THE FIELD OF CONSIDERATION NONPARAMETERIZATION model – D–DIMENSION 35
3.5 SUMMARY 42
4.1 INTRODUCTION 45
4.2 PARAMETERIZATION model 48
4.2.1 EULER EQUATION – LANGANGER 50
4.2.2 AFFINE MODEL WITH PARAMETERIZATION 51
4.2.3 DYNAMI IMAGE MODEL 52
4.2.4 TRANSFORMATION MODEL 53
4.2.5 FIXED TRANSFORMATION MODEL 53
4.3 NUMERICAL SUMMARY 54
4.4 EXPERIMENTS RESULTS 55
4.4.1 PARAMETERS-fix SEPARABLE PARAMETERIZATION WITH SYNTHETIC EXAMPLE 62
4.4.2 GARDEN-DEGREES OF FREE-FLOW 63
4.4.3 ROAD SERIES 66
4.5 SUMMARY 68
5.1 INTRODUCTION 71
5.2 STATEMENT 73
5.2.1 MODEL STATIONARITY EQUATION FOR PARAMETERIZATION WITH LATION 75
5.3 IMPLEMENTATION IN THE MEANS OF CONDENSATION-STABILITY COMPUTATION 76
5.4 SUMMARY 79

(Total Variation) 25

3.3.4 FUNCTIONAL ON THE FIELD OF CONSIDERATION NONPARAMETERIZATION model – D–DIMENSION

4.2.2 AFFINE MODEL WITH PARAMETERIZATION
המחקר został בנהיה
פרופ' אלפרד ברוקשטיין
בפקולטה למדעי המחשב

אני מודה לפניני על התמיכה והסיוע הנWebResponse להשלמתו.
נושאים בניהו של תרוצה

תווך על מחקר

לשם מאשר חלוק של תדרישות ללבת

התואר דוקטור לפילוסופיה

อล ביר

הוגש לسمع התכניון - מכון טכנולוגי לישראל

דצמבר 2006

הופה
נושאים בניה של תגרה

טל ניר