HAL
open science

# Image matching for 3D reconstruction using complementary optical and geometric information

Patricio A. Galindo

## To cite this version:

Patricio A. Galindo. Image matching for 3D reconstruction using complementary optical and geometric information. Other [cs.OH]. Université de Lorraine, 2015. English. NNT : 2015LORR0007 . tel-01751407

**HAL Id: tel-01751407**

**https://hal.univ-lorraine.fr/tel-01751407**

Submitted on 29 Mar 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE LORRAINE

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

**Département de formation doctorale en informatique** **École doctorale IAEM Lorraine**
**UFR STMIA**

# Image matching for 3D reconstruction using complementary optical and geometric information

Appariement d'images pour la reconstruction 3D par complémentarité optique et géométrique

***THÉSE***

présentée et soutenue publiquement le 20 Janvier 2015
pour l'obtention du

**Doctorat de l'université de Lorraine**

(Spécialité informatique)

par

Patricio A. Galindo

**Composition du jury:**

Rapporteurs: *Dr. Juho Kannala, Academy Research Fellow, University of Oulu, Oulu - Finland*
*Dr. Maxime Lhuillier, Chargé de Recherche, Institut Pascal, Aubière - France*

Président: *Dre. Dominique Bechmann, Professeure, Université de Strasbourg, Strasbourg - France*

Examinateurs: *Dr. Rhaleb Zayer, Chargé de Recherche, INRIA, Nancy - France*
*Dr. Bruno Lévy, Directeur de Recherche, INRIA, Nancy - France*
*Dre. Isabelle Debled-Rennesson, Professeure, Université de Lorraine, Nancy -France*

INRIA Nancy Grand Est
Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)
UMR 7503 - Campus Scientifique - BP 239 - 54506 Vandœuvre-les-Nancy Cedex

# Résumé

L'appariement d'images est un sujet de recherche central en vision par ordinateur. Il consiste à trouver des correspondances dans un ensemble d'images. Ces correspondances sont un ingrédient essentiel pour de nombreuses applications. Ces applications touchent à des disciplines variées couvrant l'estimation de mouvement dans la conduite assistée, le suivi automatique de la caméra dans l'industrie du cinéma, la télédétection dans l'arpentage géographique et la numérisation des sites du patrimoine culturel.

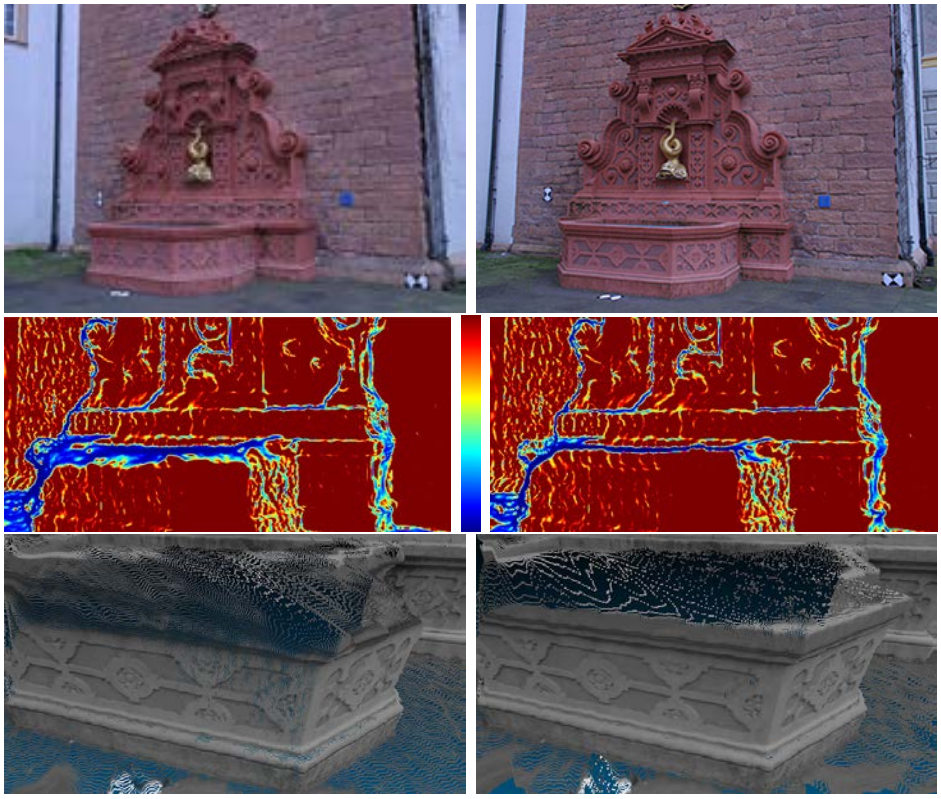
La recherche sur la problématique d'appariement d'images s'est longuement concentrée sur les aspects optiques. Mais, les aspects géométriques ont reçu beaucoup moins d'attention. L'objectif du travail entrepris dans cette thèse est l'introduction de ces aspects géométriques directement dans le problème d'appariement d'image, et ce pour les méthodes locales et globales.

Les techniques existantes pour l'appariement d'images peuvent être classées en globales et locales. Dans cette thèse, nous nous concentrons dans une première partie sur les méthodes globales basées sur le calcul des variations. Ces méthodes variationnelles peuvent fournir des résultats précis dans le cadre de la reconstruction 3D. Néanmoins les occultations visuelles et les arêtes vives, couramment rencontrés dans la pratique, posent des difficultés pour ces méthodes. En effet, dans de tels scénarios, le comportement de la solution dans ces régions dépend seulement de la contribution du régularisateur. Cela induit souvent de fausses correspondances qui ont tendance à déborder et, plus important encore, à affecter négativement les régions avoisinantes.

A l'aide d'une caractérisation géométrique de ce comportement, nous formulons une méthode d'appariement variationnelle qui dirige les lignes de la grille loin des régions problématiques, réduisant ainsi les la taille des zones affectées négativement de façon efficace et peu couteuse.

Par exemple, dans la figure ci-dessous, nous illustrons la qualité des triangles obtenus pour deux vues de la scène de la fontaine (en haut) [117]. Dans la deuxième rangée, nous montrons la qualité des triangles résultant en utilisant la méthode de correspondance variationnelle présentée dans [18] (à gauche), et en utilisant notre méthode (à droite). La couleur rouge représente triangles de bonne qualité tandis que le bleu foncé représente mauvais qualité.

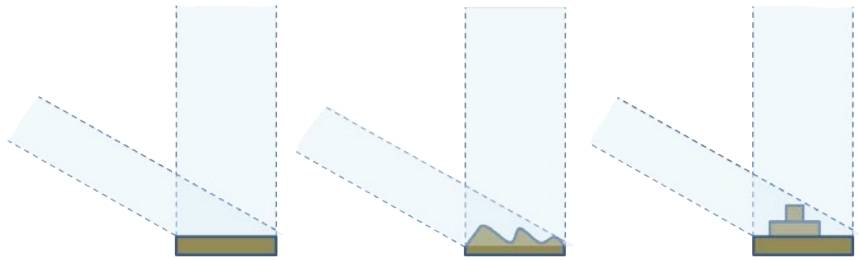
Observez l'amélioration notable, en particulier, près du rebord du bassin. Sur la rangée du bas, nous montrons un gros plan du bassin de la fontaine en utilisant les mêmes méthodes. Notez que, pour notre méthode, les lignes de la grille s'éloignent des régions problématiques.



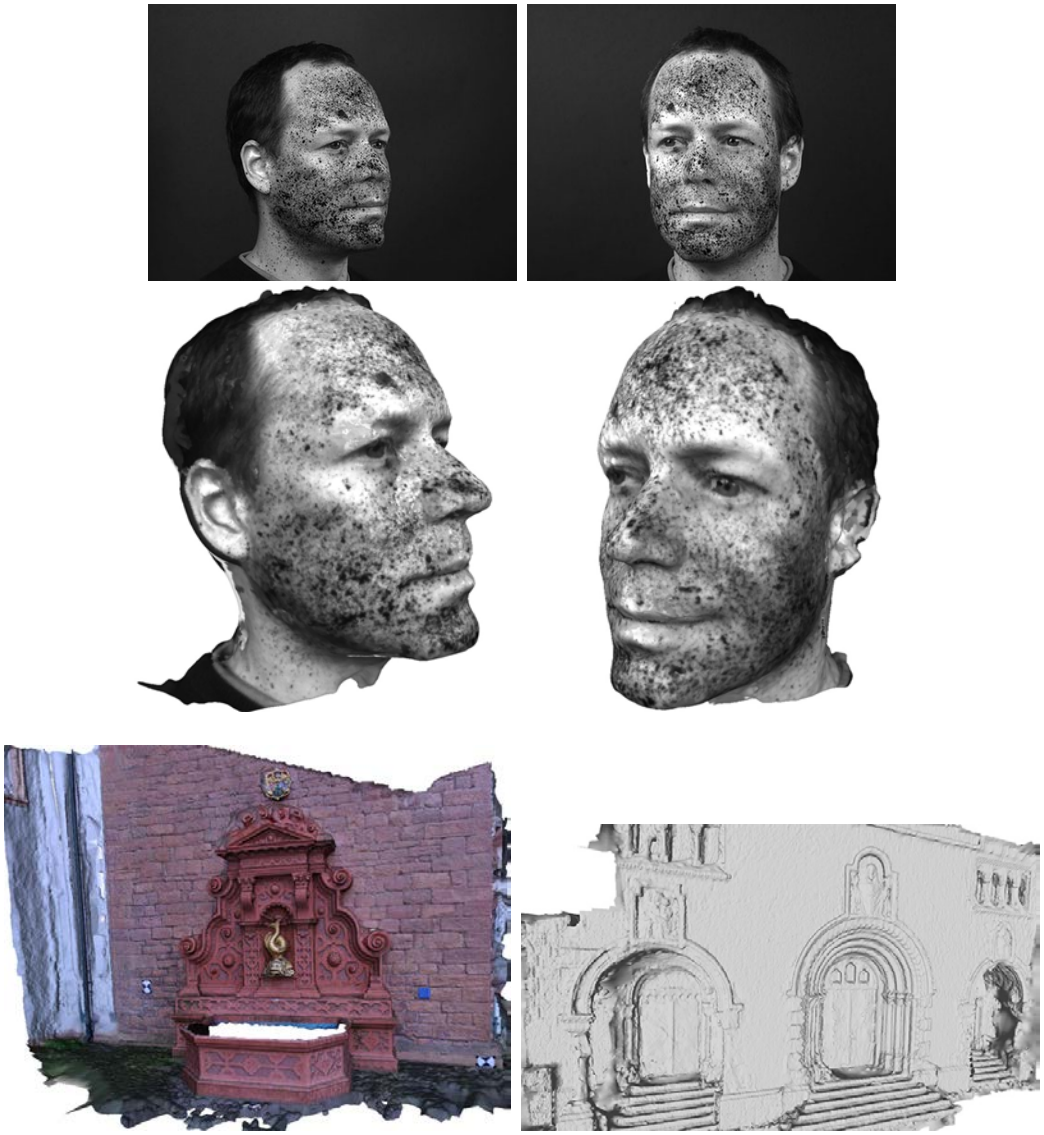
En utilisant l'approche d'appariement variationnel proposée, nous effectuons un appariement par paires d'images et ultérieurement une fusion des résultats pour obtenir une représentation complète de la scène. Cette démarche est couramment adoptée et fait souvent appel à des techniques de découpage de graphes ainsi que des tests de visibilité. Ces techniques peuvent être très gourmandes en ressources de calcul, en particulier dans les cas où des images en haute-résolution et/ou plusieurs vues sont disponibles. Par ailleurs, de petites erreurs dans l'appariement peuvent conduire à des couches superposées de surface qui ne peuvent pas être facilement traitées par des techniques standard d'élimination des données aberrantes.

Pour résoudre ces problèmes, nous gardons les appariements qui représentent le moins de distorsion entre points de vue, en profitant de la loi du cosinus de Lambert pour favoriser les contributions des zones de l'image où l'angle entre la normale à la surface et la ligne de visée est minimal (voir la figure ci-dessous). Ceci est réalisé en évaluant d'abord la mise en correspondance des cartes en 2D et en gardant les points appariés dans la plupart des points de vue, puis en donnant la préférence aux régions qui souffrent le moins de distorsion.

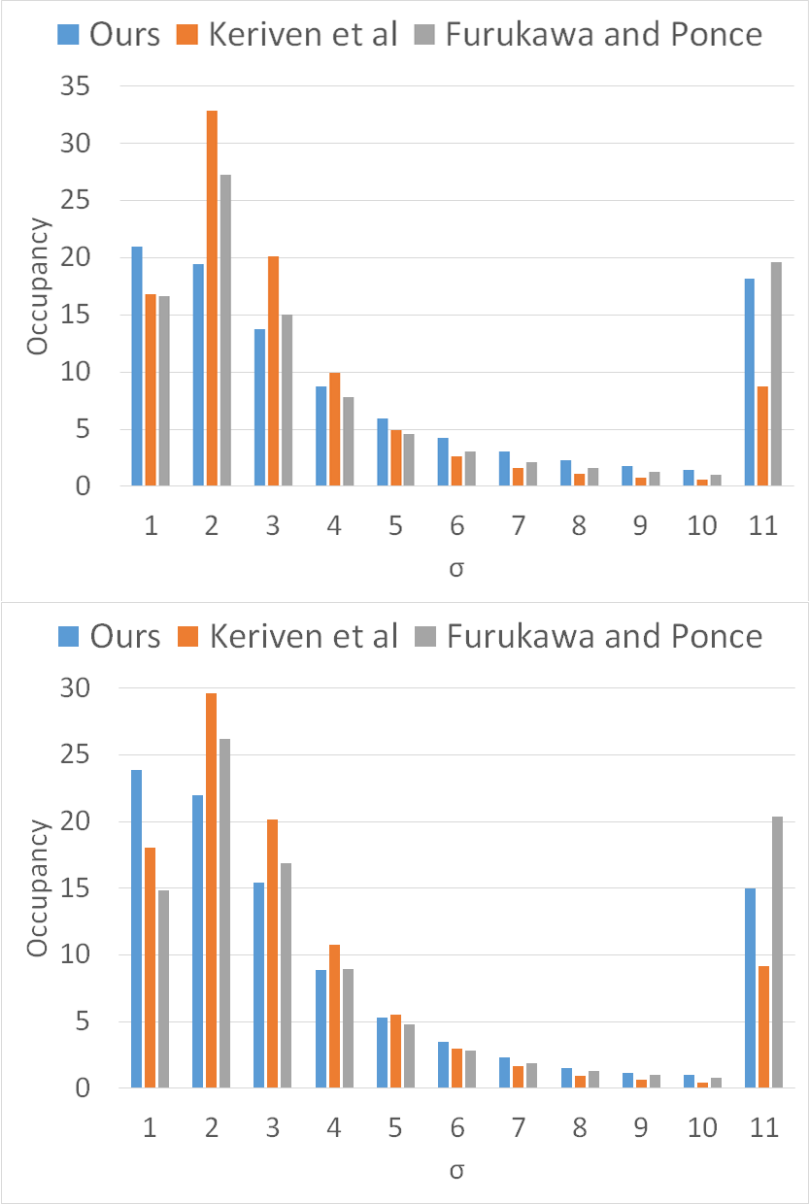
Dans la figure suivante, nous illustrons les changement de visibilité entre deux points de vue différents dans le cas d'une surface plane (à gauche). Cet effet est plus prononcé pour les surfaces lisses avec changement visible de courbure (au milieu), et les surfaces non lisses (à droite).



Dans les figures ci-dessous, nous illustrons des résultats typiques de notre méthode sur plusieurs types des données: Sam-Face (en haut), Fontaine-P11 [117] (en bas à gauche) et Herz-Jesu-P8 [117] (en bas à droite).



La figure suivante montre un histogramme de l'erreur accumulée (pour tous les points de vue) en utilisant notre méthode, [34] et [48]. Les résultats pour Herz-Jesu-P8 sont présentés en haut, et pour Fontaine-P11 en bas.



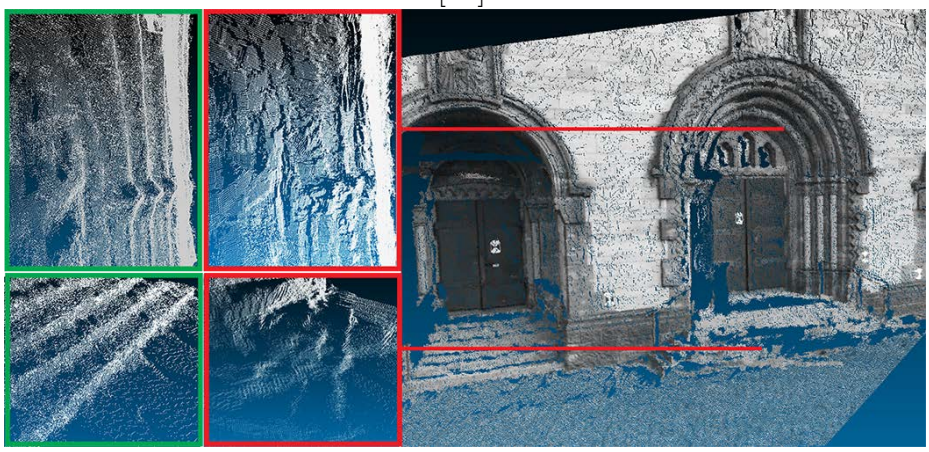
Bien que les méthodes variationnelles fournissent des résultats qui se comportent bien en général, grâce à l'équipartition des erreurs, les méthodes locales basées sur la propagation de correspondances fournissent des résultats qui s'adaptent plus étroitement à diverses structures 3D car elles n'utilisent pas de termes de régularisation. Cela peut être un avantage lors de l'appariement de régions à différentes échelles et quand des pixels voisins correspondent à des positions spatiales très différentes; ce dernier cas est souvent rencontré lorsqu'il y a de grandes discontinuités de profondeur.

D'autre part, les méthodes basées sur la propagation d'appariements ont un inconvénient important que nous illustrons dans cette thèse: des discontinuités artificielles dans les résultats obtenus du fait de la seule utilisation de l'information optique pour effectuer des estimations d' appariement.

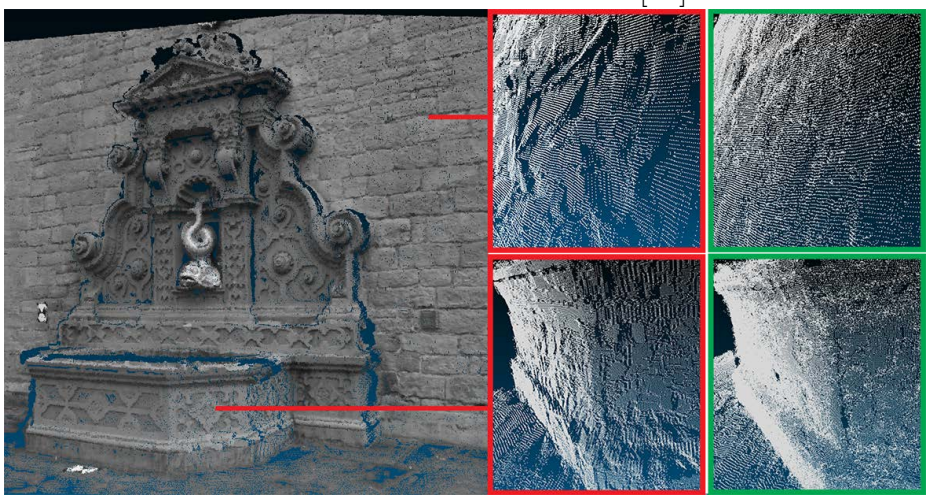
Pour surmonter ce problème, nous présentons une nouvelle façon de propager les correspondances en utilisant des informations locales sur la régularité des surfaces. De façon plus détaillée, nous commençons par la propagation des correspondances avec caractéristiques covariantes affines, et construisons au passage le nuage de points résultant. Ce faisant, nous proposons d'utiliser, pendant cette propagation, des reconstructions locales de surface pour corriger les positions des points du nuage. Cette correction en 3D modifie implicitement les correspondances et la surface, et fournit les moyens d'améliorer les estimations de transformation affines qui permettent le calcul de meilleurs appariements 2D.

Dans la figue suivante, nous montrons un exemple typique des améliorations obtenues en utilisant nos méthodes (marqué en vert). Tout d'abord, dans la scène Herz-Jesu [117] (en haut). Observez que les escaliers sont correctement reconstruit avec notre méthode (à gauche). Dans la scène de la fontaine (en bas), la qualité des résultats est nettement améliorée avect notre méthode (à droite).

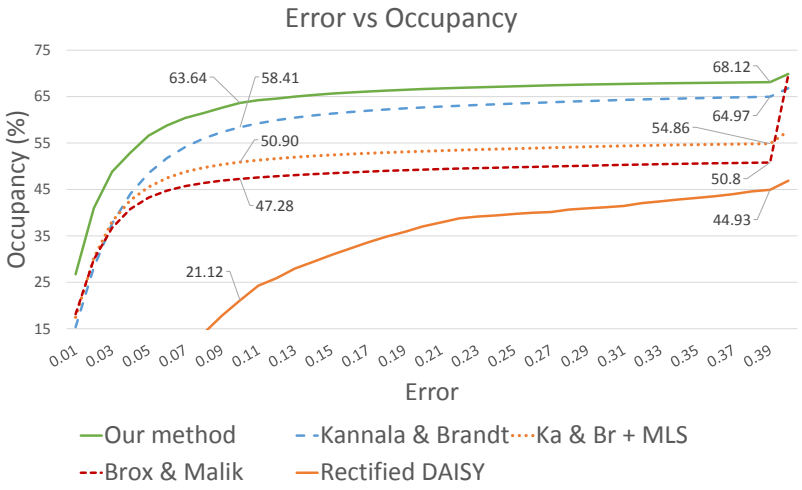
Nos résultats — Résultats avec [54]

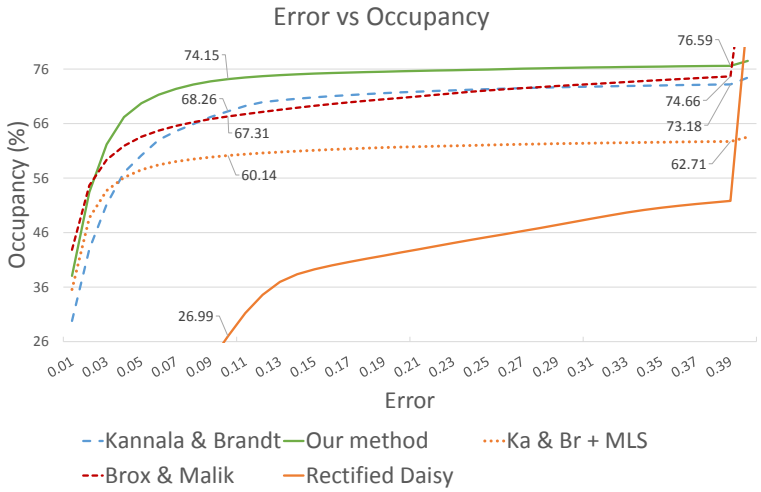


Résultats avec [54] — Nos résultats



Dans les figures ci-dessous, les graphes illustrent l'erreur de *d*epth vs occupancy de notre méthode, [54], [19], [124] (rectifié) et [54] avec MLS comme post-traitement. Les résultats pour la scène Herz-Jesu-P8 sont affichées en premier et les résultats pour la Fontaine-P11 en deuxième.



Nous évaluons notre procedure variationnelle en deux étapes (appariement et la fusion) et notre correspondance par propagation sur un banc d'essai standard ainsi que sur nos propres données. Nous montrons que la nouvelle régularisation pour l'appariement variationnel avec notre technique de fusion multi-vue, fournit des résultats plus précis que l'état de l'art.

Les gains en précision atteints grâce aux méthodes proposées, sont mesurés par des évaluations numériques par comparaison aux données réelles et sont aussi rendues probantes par une simple inspection visuelle.

# Abstract

Image matching is a central research topic in computer vision. It consists in finding correspondences across a set of images. These correspondences are a key ingredient for a variety of applications. These applications touch upon varying disciplines covering motion estimation in assisted driving, automatic camera tracking in the feature film industry, remote sensing in geographic surveying and digitization of cultural heritage sites.

Optical aspects of the matching problem have been the focus of research effort on the subject. On the other hand geometric aspects received far less attention. The aim of the work presented herein consists in the direct use of geometry in 2D matching affecting local and global methods.

The existing body of work on matching can be categorized into global and local methods. Global methods formulate the problem as a global minimization of the variation of a certain measure across images (e.g. intensities difference). In this thesis we focus on global methods based on the calculus of variations, henceforth variational matching. These methods can provide overall accurate and well-behaved results within the context of 3D reconstruction. Nevertheless, issues such as occlusions and sharp features commonly encountered in practice, raise difficult challenges for these methods. In fact, in such scenarios only the contribution of the smoothing regularizer accounts for results in those regions. This often induces wrong matches which tend to bleed into neighboring areas and, more importantly, distort nearby features. Based on a geometric characterization of this behaviour, we formulate a variational matching method that steers grid lines away from problematic regions, effectively minimizing the areas negatively affected.

Using the proposed variational matching approach we perform pairwise matching and later merging of the results to obtain a full scene representation. This is the commonly adopted approach in the literature, which is often solved using graph-cut techniques along with visibility tests. We argue that the aforesaid approach can be

highly demanding on resources, particularly in cases where high-resolution-images and/or several views are available. Moreover, small errors in matching can lead to overlapping surface layers which cannot be easily addressed by standard outlier removal techniques. To address these problems, we keep matches that represent the least amount of distortion across views, effectively taking advantage of Lambert's cosine law to favor contributions from image areas where the cosine angle between the surface normal and the line of sight is maximal. This is achieved by first evaluating the matching in 2D maps, keeping points matched in most views and giving preference to regions that suffer the least amount of distortion.

While variational methods provide well behaved results with equidistributed errors, local methods based on match propagation provide results that adapt closely to varying 3D structures since they do not use regularizers. This can represent an advantage when matching regions of different scales and when neighboring pixels match to very different locations; the latter being the case at large depth discontinuities. On the other hand, existing propagation based methods incorporate an important drawback that we illustrate in this thesis: a choppy nature in the obtained results due to the use of only optical information to perform matching estimations. To overcome this problem we present a new way to propagate matches using local information about surface regularity. In more detail, we start by propagating matches from affine covariant features, and constructing the resulting point-cloud along the way. While doing this, we propose to perform recurring local surface fittings to correct the positions of the resulting 3D points. This correction in 3D implicitly modifies the corresponding 2D matchings and, the fitted 3D surface patch provides the means to compute affine transformation estimations that allow to correspond patches in 2D.

We evaluate our variational two-step pipeline (matching and merging) and our propagation-based matching on a test-bed of standard benchmarks as well as our own datasets. We show that the new regularizer for variational matching along with our multi-view merging technique, provides more accurate results than state of the

art variational methods.

In our propagation-based matching, the corrections and new estimations based on surface fitting lead to more accurate and complete propagations.

In both settings, the gains in accuracy achieved through our methods are substantiated by numerical evaluations against ground truth data and are distinguishable by visual inspection.

To my wife and parents

# Contents

**1 Introduction** **1**

1.1 Multi-view stereo for 3D reconstruction . . . . . 2

1.1.1 Classification of Image-based 3D reconstruction methods . . . 4

1.2 Main contributions and thesis outline . . . . . 6

1.2.1 Main contributions . . . . . 6

1.2.2 Thesis outline . . . . . 7

**2 Image matching for 3D reconstruction** **9**

2.1 Introduction . . . . . 9

2.2 Local methods . . . . . 11

2.3 Global methods . . . . . 13

2.3.1 Variational techniques for global methods . . . . . 14

2.3.2 Discrete techniques for global methods . . . . . 15

2.3.3 Evaluation data-sets . . . . . 16

**3 Feature detection and matching** **17**

3.1 What are local features? . . . . . 17

3.2 Feature detection . . . . . 18

3.2.1 Curvature based feature detectors . . . . . 19

3.2.2 Intensity based feature detectors . . . . . 23

3.2.3 Segmentation based features . . . . . 28

3.2.4 Model based feature detectors . . . . . 29
3.2.5 Color based feature detectors . . . . . 29
3.3 Feature descriptors and matching . . . . . 30
3.3.1 Feature descriptors based on distribution . . . . . 30
3.3.2 Feature descriptors based on filters . . . . . 31
3.3.3 LDAHash . . . . . 32
3.3.4 Feature descriptors based on spin images . . . . . 32
3.3.5 Feature descriptors based on color . . . . . 32
3.4 3D reconstruction using image features . . . . . 33
3.4.1 DAISY . . . . . 33
3.4.2 SIFT Flow . . . . . 34
3.5 Experiments . . . . . 34
3.6 Conclusion . . . . . 35

**4 Variational Image Matching for 3D reconstruction 43**
4.1 Classical formulation . . . . . 44
4.2 Data term . . . . . 45
4.2.1 Data term penalizers . . . . . 45
4.2.2 What to measure . . . . . 47
4.3 Regularization . . . . . 50
4.4 Data term linearization . . . . . 51
4.4.1 Early data term linearization . . . . . 52
4.4.2 Late data term linearization . . . . . 52
4.5 Additional cues from scene geometry . . . . . 54
4.5.1 Epipolar geometry . . . . . 54
4.5.2 Sparse feature matches . . . . . 56
4.6 Solving the variational matching problem . . . . . 57
4.6.1 Successive over relaxation: SOR . . . . . 58
4.6.2 Alternating direction implicit: ADI . . . . . 59

4.6.3 Coarse to fine . . . . . 60
4.6.4 Multi-grid . . . . . 61
4.7 Experiments . . . . . 62
4.7.1 Datasets . . . . . 63
4.7.2 Methods and criteria . . . . . 65

**5 Distortion driven variational multi-view reconstruction 77**
5.1 Introduction . . . . . 78
5.2 Geometry driven variational matching . . . . . 80
5.2.1 Classical variational matching . . . . . 80
5.2.2 Distortion driven variational matching . . . . . 82
5.2.3 Results using distortion driven variational matching . . . . . 84
5.3 Distortion driven multiple-view merging . . . . . 86
5.3.1 Triplet contributions . . . . . 87
5.3.2 Pair contribution . . . . . 89
5.4 Results and discussion . . . . . 90
5.5 Conclusion . . . . . 93

**6 Propagation-based matching for 3D reconstruction 99**
6.1 Introduction . . . . . 99
6.2 Match propagation for wide-baseline configurations . . . . . 101
6.3 Quasi-dense match propagation for wide-baseline configurations . . . 105
6.3.1 Quasi-Dense Wide Baseline Matching for Three-Views . . . . 107
6.3.2 Multi-view quasi-dense matching for wide-baseline configurations 108
6.3.3 Results and discussion . . . . . 109
6.4 Accurate, Dense, and Robust Multiview Stereopsis . . . . . 111
6.4.1 Results and discussion . . . . . 114

**7 Complementary geometric and optical information for match propagation based 3D reconstruction** **119**
7.1 Introduction . . . . . 120
7.2 Geometry based image match propagation . . . . . 121
7.2.1 Overview . . . . . 121
7.2.2 Propagation . . . . . 123
7.2.3 Candidate region querying . . . . . 126
7.2.4 Fit surface and update . . . . . 127
7.3 Results . . . . . 130
7.4 Conclusion and discussion . . . . . 131

**8 Conclusion and perspective** **141**
8.1 Conclusion . . . . . 141
8.2 Perspective . . . . . 143

**Publications** **146**

**Bibliography** **147**

# List of Figures


1-1 Triangulation for 3D reconstruction. Once correspondences are found (left figure) and having all the information about the cameras ($V_1$ and $V_2$), the 3D structure of the scene can be triangulated (right figure) . 3

2-1 On the left: apparent object motion caused by point of view motion. On the right: apparent motion caused by actual motion of the observed object . . . . . 10

3-1 Example of the Harris [45] corner detector for increasing sensitivity values. From top to bottom, values taken by $\kappa$ are 0.001, 0.1 and 0.2. This example shows that for low values of $\kappa$ even small curvatures are detected as corners. . . . . 21

3-2 Example of the Harris [45] corner detector for increasing sensitivity values. From top to bottom, values taken by $\kappa$ are 0.3, 0.7 and 0.9. This example shows that as we increase the value of $\kappa$ only sharp corners are detected. . . . . 22

3-3 Geometric representation of equation 3.8. . . . . 26

3-4 SURF filter approximations. Top row, discretized Gaussian second order partial derivatives ($G_{yy}$,$G_{xx}$ and $G_{xy}$). Bottom row, corresponding SURF box filter approximations. . . . . 27

3-5 Intensity-based region detection. Rays emanating from $I_0$ find a maximum at the boundaries of the region (green dots). The dotted (red) ellipsoid represents the affine transformation used for later normalization. 28
3-6 Illustration of the grid used to compute the SIFT descriptor. The green dot represents the location of the feature detected. The cell with the arrows represents a histogram with the eight bins. In the lower right cell is exemplified a case for $6 \times 6$ pixels. . . . . 31
3-7 Fountain scene, views $0 - 1$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red . . . . . 36
3-8 Fountain scene, views $0 - 2$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red . . . . . 37
3-9 Fountain scene, wide-baseline pair of views $0 - 4$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red 38
3-10 Zoom out example for the painting dataset, views $0 - 1$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red . . . . . 39
3-11 Rotation example for the painting dataset, views $0 - 2$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red . . . . . 40
3-12 Slanting, rotation and scaling for the painting dataset, views $0 - 3$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red . . . . . 41

4-1 To the left, shape of the penalizers $f(s^2) = s^2$ (in blue) and $f(s^2) = \sqrt{s^2 + \epsilon^2}$ (in red) for $s = [-2, 2]$. To the right, influence of the square penalizer given by $f'(s^2) = s$ (in blue) and influence of the robust penalizer given by $f'(s^2) = s/\sqrt{s^2 + \epsilon^2}$ (in red) . . . . . 46
4-2 Epipolar geometry . . . . . 55
4-3 Fountain scene. Views 2 and 7 . . . . . 63

4-4 Dino scene. Views 26 and 31 . . . . . 64
4-5 Face scene. Views 1 and 6 . . . . . 64
4-6 Box 1 with repetitive pattern . Views 1 and 6 . . . . . 64
4-7 Endpoint error results for the fountain scene, view 2-5. From left to right and top to bottom; Brox04, Brox11, Sun, Liu, Wedel, Brox04+EF, Kannala, Furukawa. On the bottom, the error color map with dark red representing distances of one hundred pixels or more . . . . . 68
4-8 Percentage of endpoint error less or equal to 1 pixel in the fountain scene 69
4-9 Average endpoint error in the fountain scene . . . . . 69
4-10 Average distance to features in the fountain scene . . . . . 70
4-11 Average distance to the epipolar line in the fountain scene . . . . . 70
4-12 Average distance to features in the dino scene . . . . . 71
4-13 Average distance to the epipolar line in the dino scene . . . . . 72
4-14 Distance to the epipolar line results for the face scene, views 1-2. From left to right and top to bottom: Brox04, Brox11, Sun, Liu, Wedel, Brox+E+F, Kannala, Furukawa. On the bottom, the error color map with dark red representing distances of three pixels or more . . . . . 73
4-15 ADF in the face scene . . . . . 74
4-16 ADE in the face scene . . . . . 74
4-17 Average distance to features in the box scene . . . . . 75
4-18 Average distance to the epipolar line in the box scene . . . . . 75

5-1 A zoom on the reconstruction results of classical variational matching (bottom) across a pair of views from the Fountain-P11 [120] and our face1 data sets (top). Please note, the deterioration along the rim of the fountain and the tip of the nose. . . . . 81

5-2 Two slices through the fountain back-wall using the classical formulation of variational multi-view reconstruction. Besides from the overwhelming data size, slight matching errors can lead to over- (left) and inter- (right) laying data. . . . . 81

5-3 Color coded visualization of the quality of reconstructed triangles (projected into the image plane), using the classical formulation (top-left), and our proposed method (top-right). Red and blue represent best and worst quality resp. Closeups to reconstructions of the fountain's base are shown at the bottom. Please note the crisper rim to the right. . . 85

5-4 Highlighted region on the rim of the fountain (left) and closeups on the results using our method (center) and using Brox et al. [18](right). Our results show a correctly reconstructed rim which displays regular triangles. . . . . 85

5-5 Illustration of the area change across two different views in the case of a flat surface (left). This effect becomes more pronounced for smooth surfaces with visible curvature change (middle), and non-smooth surfaces (right). . . . . 86

5-6 Mapping of a (triangulated) regular grid (left) into a deformed configuration. Visualization of the deformation by means of its action on a circle (right). The elongation along the principal axis represents the square roots of the eigenvalues of the transformation. . . . . 87

5-7 Triplet matches are validated based on trifocal transfer within the triplet (top). The selection of best viewed regions in a triplet is performed by transferring the grid to neighboring views and analyzing its distortion (bottom). . . . . 89

5-8 Pair matches are validated by epipolar distance and a forward-backward map. The selection of best viewed regions in a pair is performed by transferring the grid to neighboring views and analyzing its distortion (bottom). Only regions which where not covered by triplets are considered. . . . . 90

5-9 Processing best viewed regions in the Fountain data set [120]. Each view represents the central image of a triplet (other two images not shown). The red-colored regions represent areas best viewed in the triplet. Yellow regions represent regions that are only visible in the triplet and therefore are included even if they do not comply with the best view requirement. . . . . 91

5-10 Top row, result of our approach on the Fountain-P11 data set [120] and a histogram of the error accumulated for all the views obtained with our method, [34] and [48]. Bottom, distance error for the depth of view five: ours, [34] and [48]. The red color represents no result or result farther than $30\sigma$, green represents the regions where no result can be obtained, dark gray represents larger error (smaller than $30\sigma$). 92

5-11 Top row, result of our approach on the Herz-Jesu-P8 data set [120] and a histogram of the error accumulated for all the views obtained with our method, [34] and [48]. Bottom, distance error for the depth results of view five: ours, [34] and [48]. The red color represents no result or result farther than $30\sigma$, green represents the regions where no result can be obtained, dark gray represents larger error (smaller than $30\sigma$). 93

5-12 Two images from face2 dataset-$6 \times 1.3$ MP- (top) reconstructed using our approach (middle) and using variational matching code provided by the authors of [18]. . . . . 95

5-13 Two images from face1 dataset-9 $\times$ 8 MP- (top) reconstructed using our approach (middle) and without the use of our distortion driven matching. . . . . 96

5-14 Two images from our boot dataset-5 $\times$ 4 MP- (left) reconstructed using our approach (middle) and without the use of our distortion driven matching (right). . . . . 97

6-1 Window correspondences from narrow to wide baselines (going from left to right and top to bottom). This figure illustrates how rectangular windows (in green) around corresponding points, do not faithfully represent corresponding regions. On the other hand, windows adapted through an affine transformation (in blue) clearly approximate better the surface at hand. . . . . 103

6-2 Illustration of the sidedness verification across two views. Regions $R_1, R_2, R_3$ pass the verification since, in both images, $R_2$ belongs to the left side of the vector that connects $R_3$ to $R_1$. For the case of the regions $R_4, R_5, R_6$ the sidedness check fails since $R_5$ changes sides across the vector that connects $R_6$ to $R_4$. . . . . 104

6-3 Illustration of the effect of choosing a magnifying affine transformation instead of a reducing one. Left column, two views of a scene with matching points denoted by a connecting red line. In the middle column, corresponding windows (around matching points) using a reducing transformation that achieves a correlation score of 0.69. On the right column, corresponding windows (around same matching points as in the previous example) using a magnifying transformation that achieves a correlation score of 0.83 . . . . . 107

6-4 Depth error comparison for the Fountain-P11 scene. In the top row, the views two and three used to reconstruct the scene (left) and a cumulative depth error chart for the current views using methods [64], [54] and [34]. In the bottom row, color coded depth error results using methods [64] (left) and [54] (right). Color code bar (center), where dark blue represents errors close to zero and dark red represents errors equal or greater to 0.15. . . . . 110

6-5 Depth error comparison for the Fountain-P11 scene for a case of a wider baseline. In the top row, the views two and four used to reconstruct the scene (left) and a cumulative depth error chart for the current views using methods [64], [54] and [34]. In the bottom row, color coded depth error results using methods [64] (left) and [54] (right). Color code bar (center), where dark blue represents errors close to zero and dark red represents errors equal or greater to 0.15. . . . . 111

6-6 Depth error comparison for the Herz-Jesu-P8 scene. In the top row, the views three and four used to reconstruct the scene (left) and a cumulative depth error chart for the current views using methods [64], [54] and [34]. In the bottom row, color coded depth error results using methods [64] (left) and [54] (right). Color code bar (center), where dark blue represents errors close to zero and dark red represents errors equal or greater to 0.15. . . . . 112

6-7 Two of the three filters used in [35]. **On the left**, first filter that verifies neighboring matches and keeps points closer to the cameras. **On the right**, a common visibility computation using a depth map test. In this case, the point $p$ marked by a red cross does not pass a visibility test. . . . . 114

6-8 Example of a reconstruction using [34]. Top row, four $1,944 \times 1,296$ pixels images used for reconstruction. Second row, two views of a point cloud obtained using [34]. Bottom row, two views of a meshed version of the upper pointcloud using [55] . . . . . 116

6-9 Example of a reconstruction using [34] for the Fountain-P11 dataset. Top row, two views of a point cloud obtained using [34]. Bottom row, two views of a meshed version of the upper pointclouds using [55] . . 117

7-1 Reconstruction from views 3-4 of the Herz-Jesu P8 dataset using [54] (right). Geometric artifacts are clearly visible in the closeups (middle). Whereas, the geometry of the scene is better captured with our approach (left). . . . . 121

7-2 Reconstruction from views 2-3 of the Fountain P11 dataset using [54] (left). Geometric artifacts are clearly visible in the closeups (middle). Whereas, the geometry of the scene is better captured with our approach (right). . . . . 122

7-3 Illustration of a few propagation steps before performing any geometric updates. . . . . 123

7-4 Illustration of the geometric fitting steps. First, the candidate region is selected (a), the support and the core are colored in green and blue respectively. Second, local surface fitting is performed (b). Third, the core points are projected onto the surface (c). . . . . 124

7-5 Searching for best region to fit a surface patch. Confirmed points are depicted in blue, unconfirmed ones in red. Points in the orange region were obtained in the preceding propagation step. In green, support (outer) and core (inner) windows. . . . . 127

7-6 Results of our approach on the Fountain P11 views 2-3 (top-left). On the top-right, a chart illustrating the depth error vs occupancy of our algorithm, [54], [19], [124] (rectified) and [54] with MLS as post-processing.On the bottom color coded depth error using our algorithm (left) and using [54] (right). . . . . 132

7-7 Closeups on portions of the wall and fountain-top for the Fountain P11 views 2-3 using [54] (top) and using our method (bottom) . . . . . 133

7-8 Result of our approach on the Herz-Jesu P8 views 3-4 (top-left). On the top-right, a chart illustrating the depth error vs occupancy of our algorithm, [54], [19], [124] (rectified) and [54] with MLS as post-processing. On the bottom, color coded depth error using our algorithm (left) and using [54] (right). . . . . 134

7-9 Closeups on to results on Herz-Jesu P8 views 3-4 using [54] (top) and using our method (bottom). A closeup to a portion of the main stairs (left) and to the left-most stairs in the scene (right) . . . . . 135

7-10 Results on two views of our face dataset. The two views used (top), our result with a closeup to the cheek (middle), results using [54] (bottom) 136

7-11 Results on views 7 and 15 of the warrior dataset from [33]. The two views used are showed at the top row along with the 7 features matched. On the bottom row, our result (left) and results using [54] (right). Notice that our method returns less holes and it is able to reconstruct the warrior's hammer. . . . . 137

7-12 Comparisons of our results to those of DAISY [124]. Results on the Fountain scene (top) and the Herz-Jesu scene (bottom) for pairs of images with increasing baselines. The charts present the percentage of correct depth estimations for each pair of views. A depth estimation is considered correct if it presents and error of less than 1% of the scene's depth range [124] when compared to the laser scanned data. . . . . 138

7-13 Comparisons between propagation based methods and variational methods. The methods used are: our variational (explained in this chapter), the method of Kannala and Brandt [54], our variational method proposed in chapter 5 and the method of Brox and Malik [19] . . . . . 139

8-1 Color coded depth error using our algorithm (left) and using [54] (right). 144

8-2 Color coded propagation for the fountain scene where each color represent matches propagated from a seed. To the left, propagation results using the method proposed in Chapter 7. To the right, sample propagation of our envisioned method outlined in this section. . . . . 145

# List of Tables

4.1 Average end point error in the fountain scene . . . . . 68
4.2 Average distance to the epipolar line in the fountain scene . . . . . 71
4.3 Average distance to features in the fountain scene . . . . . 71
4.4 Average distance to the epipolar line for the dino scene . . . . . 72
4.5 Average distance to features for the dino scene . . . . . 72
4.6 Average distance to the epipolar line for the face scene . . . . . 73
4.7 Average distance to features for the face scene . . . . . 74
4.8 Average distance to the epipolar line for the box scene . . . . . 76
4.9 Average distance to features for the box scene . . . . . 76

8.1 Pros and cons of variational and propagation based $3D$ reconstruction 143

# Chapter 1

# Introduction

Computer vision aims to provide methods for obtaining relevant information from images. Since its inception it has been inspired by research on human perception. In general, computer vision attempts to emulate or augment human perception. For example, it emulates certain human tasks such as scene and object classification or face recognition. It augments it by performing accurate scene reconstructions or visual odometry. Nowadays, these computer vision methods affect of our every day life, for example, through visual surveillance, medical imaging, industrial inspection and human-computer interaction.

An important vision-related task that we humans perform constantly and naturally is that of perceiving depth. This task is called stereopsis and it works by combining the information provided by our two eyes (binocular vision) and prior knowledge about scenes, objects and lightning. The stereopsis process, and its importance, have been well understood for quite some time now. Going back in history, during World War II an Anglo-American campaign named Crossbow (originally Bodyline) aimed to find German long range weapons, mostly by obtaining images from planes sweeping German-controlled regions [56]. The photographs obtained were vast, making the task of (manual) inspection a tedious one. To overcome this problem, they developed

a technique to visualize the images in 3D using special glasses. This was accomplished by taking overlapping pictures of the fields and visualizing them in pairs with 3D glasses similar to the ones used in movie theaters nowadays.

Since then, computer vision solutions to the stereopsis problem have evolved to great extent, making use of direct visual input and further priors. A prior commonly used is surface smoothness, where objects in the scene are assumed to be piecewise smooth. In general, results can be greatly improved by using the right priors at right locations. In this thesis, we will propose to use local geometry information to improve on results obtained mainly from visual input. We start by observing, in the next section, the key elements of matching for 3D reconstruction.

## 1.1 Multi-view stereo for 3D reconstruction

Using common (pinhole) cameras the image acquisition process consists in projecting 3D points towards the center of the cameras. The points at which these projection lines intersect the image planes construct images as we know them. In this thesis, we are interested in the inverse problem, which essentially aims to find corresponding points in multiple 2D images to triangulate the position of the 3D point. For example in figure 1-1, once correspondences are found (left), the 3D positions can be triangulated (right).

The pinhole camera [46] can be decomposed in the following manner:

$$P = K[R|t] \tag{1.1}$$

with $P$ the $[3 \times 4]$ camera matrix that represents the relationship between a $3D$ point and it's projection into a $2D$ image plane following a pinhole camera model. $K$ the intrinsic camera parameters, $R$ a rotation matrix, $t$ a translation vector that corresponds to $t = -RC$ where $C$ is the center of the camera.

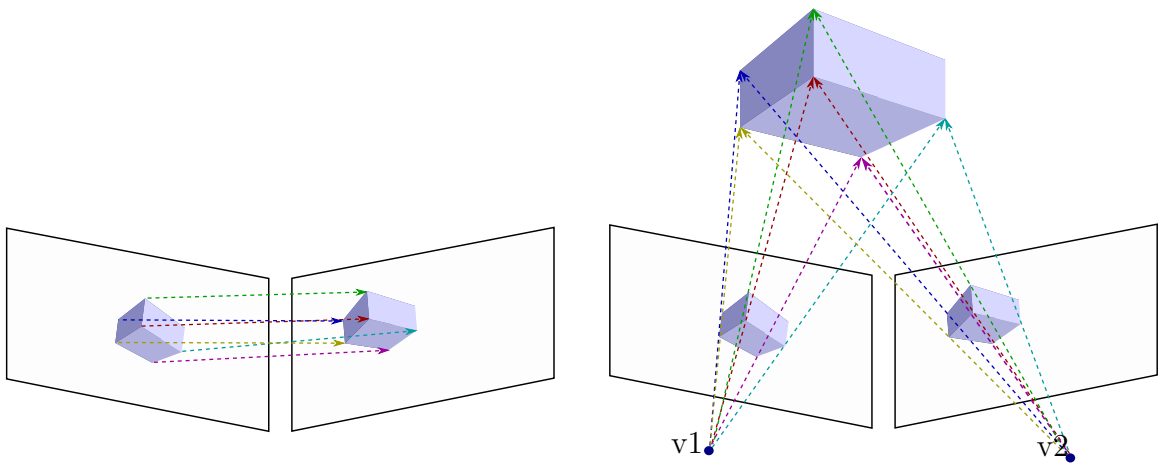



Figure 1-1: Triangulation for 3D reconstruction. Once correspondences are found (left figure) and having all the information about the cameras ($V_1$ and $V_2$), the 3D structure of the scene can be triangulated (right figure)

The intrinsic matrix $K$ consists of the following parameters:

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \tag{1.2}$$

with $\alpha_x = fm_x$ and $\alpha_y = fm_y$ stand for the focal length of the camera in terms of pixel dimensions in the $x$ and $y$ directions. $s$ stands for the skew (usually zero) and $[x_0, y_0]$ represents the principal point in pixel dimensions.

The camera parameters can be obtained in an automatic manner following one of several proposed methods (e.g. [73]). In general, such methods start by matching a sparse set of features or patterns in several views and solving a non-linear multivariate system of equations. In this process, RANSAC fitting is used to remove wrong matches [73]. If the calibration is done using an object for which real dimensions are known (e.g. [115]) or if some of the parameters are known in advance (like translation e.g. [123]), one can recover a metric camera projection. In other cases, one can only

recover camera projections that allow scene representation up to a scale factor.

### 1.1.1 Classification of Image-based 3D reconstruction methods

Roughly four classes of methods exist for the task of multi-view image-based 3D reconstruction [111]. These methods are attached to five different ways of scene representation: point-clouds, meshes, depth-maps, level-sets or voxels.

The first class of reconstruction methods aims to obtain a set of 3D points (point-cloud) to which a surface can be fitted to represent the final result (e.g. [88, 34]). For example, a widely used surface fitting method is the Poisson Surface Reconstruction [55] that receives a point-cloud and returns a polygonal mesh. The surface fitting step can be done simply to provide a final representation of the scene that can be easily visualized. It can also be done as an intermediate step needed to further optimize the 3D results using the notions of surface and connectivity (e.g. [35]). In this class of methods, two main scene representations are used: point-clouds and polygonal meshes. A point-clouds consists of a set of points defined by their coordinates (X,Y,Z in 3D). These set of points can be accompanied by further information such as color and normal. Polygonal meshes represent the objects in the scene by a set of (connected) planar facets; where triangles are one of the most used elements. Arguably, this is the most used representation since it is efficient regarding storage space and it can directly profit from a vast set of available geometric tools.

The second class of reconstruction methods aim to compute the observable depth for each point (pixel) in an image, essentially computing depth maps. This methods either solve independently for several views and then merge the results as last step (e.g. [91]), or solve ensuring multi-view consistency at all times [119, 37]. Depth-maps store the depth value for every pixel in an image but, in general several depth maps are needed to represent a whole scene. This representation is useful for tasks such as autonomous vehicles navigation (e.g. [63]).

Another way to attack the problem in hand is to obtain an initial estimation of the scene and evolve it until a fair result is obtained. This can be done, for example, by representing the scene using voxels and use space carving techniques (e.g. [62]). Other methods include the use of level-sets and mesh-based methods, that move the vertices of the mesh to minimize an energy function [101]. Level-sets and voxels are representations based on a 3D grid. For level-sets, each cell encodes its closest distance to the surface; usually represented by a negative value for the cells inside the object and positive value for the cells outside the object. For the voxel representation, each grid cell is marked as occupied or not (by the object). These two representations (voxels and level-sets) need to manage a tradeoff between accuracy and memory efficiency, which is defined by the size of the cells used in the 3D grid. This is not always a straightforward problem since the resolution needed is dependent on the (unknown) objects in the scene. Nevertheless, this shortcoming can be minimized by using tree-like representations (e.g. [28]).

The fourth class of reconstruction methods, works by formulating the problem as a 3D cost. The resulting volume can be represented by a polygonal mesh.

From the five forms of scene representation mainly three are used in most of the literature: point-clouds, polygonal meshes and depth-maps. Voxels and level-sets can be problematic regarding memory efficiency and accuracy since the volume of the scene needs to be discretized and the optimal resolution of this discretization is not known beforehand. In theory, it is preferred to use small voxels for small details, larger voxels for larger objects and empty spaces but, choosing this resolutions while recovering the scene structure is not a simple task.

Depth-maps have the advantage that they discretize the problem according to the number of pixels in each image and they only reconstruct what is seen on an image. The problem with depth maps is that they introduce the new task of having to merge many of them to fully represent a 3D scene [122].

Polygonal meshes on the other hand, are an efficient representation that introduces

simpler problems, in particular regarding the maintenance of data structures and geometric coherence. Furthermore, using this representation one can easily tap into an extensive pool of geometry processing tools.

Point-clouds are suitable for some applications, such as robotic navigation, but they are usually converted to polygonal meshes for solid-like visualizations and geometric computations.

This thesis works in the direction of the first class of reconstruction methods, where a point-cloud is sought. The obtained point-clouds can then be meshed, using for example [94]. Using this class of methods allows to obtain the shape and appearance of scenes, maintaining a direct relation between the 2D in-image points and the resulting 3D point-cloud. Such relation is not maintained with all methods. For example, space-carving techniques can work very fast reconstructing (mostly small) objects but the mentioned 2D-3D association is lost in the process. This relation is important because it allows the direct estimation of objects' appearances and it also enables time-related evolution of 3D models. For example, one can compute an initial 3D object and model its evolution using optical tracking techniques along with numerical simulations [113].

## 1.2 Main contributions and thesis outline

### 1.2.1 Main contributions

In this thesis we address the problem of 3D reconstruction using information about local geometry to complement optical matching. We do this in the context of global variational 3D reconstruction and in the context of local propagation-based 3D reconstruction.

Specifically, we propose two methods for matching and 3D reconstruction. These methods are motivated with the analysis of local deformations and regularity and, substantiated with examples and benchmarks. Starting from three main assumptions:

i) camera calibration is known, ii) the images are corrected for radial distortion and iii) the scene is static, we propose:

- To use local $3D$ deformation characterization to adapt and improve variational matching so large jumps in scene depth are respected and not destroyed by smoothness assumptions. A variational matching technique is formulated along with a multi-view stereo merging algorithm, to provide a complete multi-view 3D reconstruction method.

- A propagation-based 3D reconstruction algorithm that introduces spacial smoothness during propagation.

### 1.2.2 Thesis outline

This thesis is organized as follows. First, we discuss the two main families of image matching techniques: global and local. We follow by outlining these two families of methods in chapter 2. This outline comes in hand to prepare the reader for the following chapters. Next, we present feature detection and matching techniques (Chapter 3) since they represent an important subject in Multi-View stereo and general 3D reconstruction. We then present variational matching techniques (Chapter 4) along with evaluations which demonstrate the advantages of including features and scene geometry as constraints. Improving on these methods, we proposed a variational matching technique that takes into account local measurements of spatial deformation to improve matching along and around large depth jumps in the scenes (chapter 5). This matching method is coupled with a multi-stereo merging technique that prioritizes matches obtained with pairs of views with less viewing distortion. We continue by presenting a special breed of local matching methods, namely, propagation-based methods (Chapter 6). These methods are presented and evaluated in the context of 3D reconstruction and we identify regularity and completeness shortcomings. Therefore, we propose a method that overcomes these problems by taking into account the

fact that there is a surface being built, which we can assume to be smooth (Chapter 7). To this end, we propose to fit a local surface patch while propagating and project the points to the fitted surface. By doing this, we manage to preserve the versatility of this family of methods. We evaluate this proposed method, showing that it provides smooth results that are more complete and more accurate than related work. Finally, we present our general conclusion and future work in Chapter 8.1.

# Chapter 2

# Image matching for 3D reconstruction

In order to perform 3D reconstruction using multiple images one needs to find correspondences across the set of images being used. Hereon, we focus on a more basic but key problem in 3D reconstruction: finding matches across an image pair. In general, two families of image matching methods exist: local and global. The local methods work by matching local information and are known for their versatility and simplicity. Global methods formulate the matching problem as an optimization and solve for all the image pixels. They are characterized by overall smoothness and error equidistribution.

## 2.1 Introduction

Finding correspondences across multiple images is a general and essential task in computer vision. It is a fundamental stage in the acquisition of 3D scene models from multiple photographs and its solutions are closely related to those of optical flow. In this chapter we lay down the main ideas of local and global image matching methods.

Global methods pose the problem as a global optimization of the difference of some property (e.g. image intensities) along with a smoothness condition (e.g. Laplacian). The local methods solve the problem by matching small, local neighbourhoods in both images.

Matching estimation of relative displacements across images is closely related to the optical flow problem, which is represented as a vector field that describes the motion between two images. This vector field is obtained by finding the matchings between two images, posing the displacements per-pixel as unknowns. These displacements represent apparent motion of objects captured in the images; where the apparent motion can be the result of real objects displacements, real point of view (camera) displacement or both (see figure 2-1).

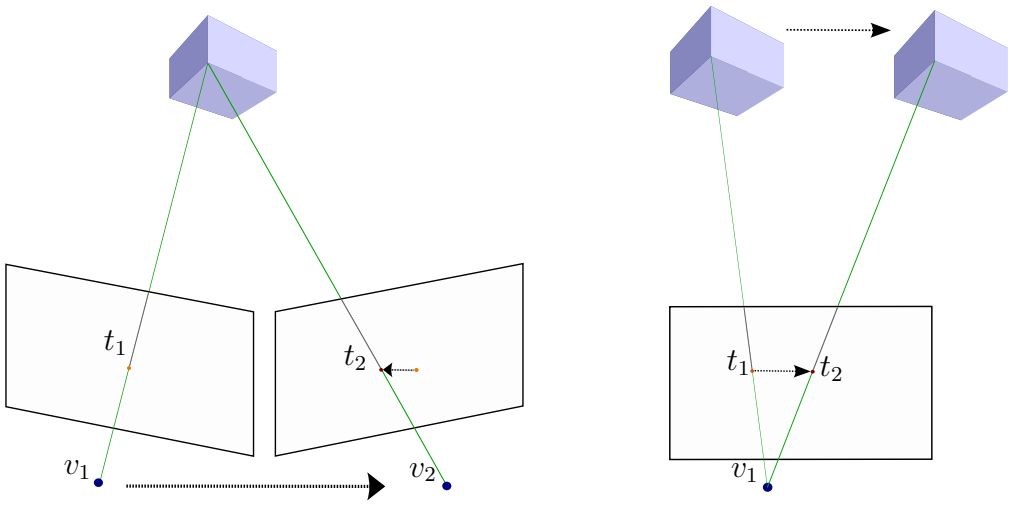



Figure 2-1: On the left: apparent object motion caused by point of view motion. On the right: apparent motion caused by actual motion of the observed object

The estimation of optical flow is analogous to the matching problem and related to more general problems such as 3D reconstruction, tracking, video stabilization, video compression and even segmentation.

In general, the objective of the optical flow estimation for two images, $I_1$ and $I_2$,

can be formulated as finding the displacements $u$ and $v$ such that:

$$I_1(x, y) = I_2(x + u, y + v) \tag{2.1}$$

where $I_1$ and $I_2$ represent to different images that share information and $(x, y)$ are image coordinates. In this case it is assumed that the image intensities (for corresponding points) do not change from image 1 to image 2. This is not always the case but it is a reasonable approximation that enables a simple formulation of the problem. In the continuing chapters, we will detail cases where this approximation fails and how to improve it. Furthermore, both local and global families of methods will be outlined in the next sub-sections.

## 2.2 Local methods

Local image matching methods aim to find correspondences across images by independently matching small image regions. These methods assume that local information surrounding a point in an image will also be present around the corresponding point in another image. This assumption is violated in various cases. For example, for corners of objects and partial occlusions in certain images. Nevertheless it is a reasonable assumption for flat and regular object areas.

In more detail, local methods for the computation of correspondences assume certain local rigidity and find correspondences using window-based image matching. For example, Lucas-Kanade [76] proposed that the optical flow is constant in a small neighborhood around a pixel and, therefore, it can be directly computed from equation 2.2; where $G$ represents a weighting window (usually Gaussian), $\rho$ represents the size of the window and the location of the central pixel is denoted by $[x, y]$.

$$E(u, v) = G_\rho * ((I_x u + I_y v + I_t)^2) \tag{2.2}$$

The minimization of 2.2 is directly computed using 2.3

$$
\begin{aligned}
G_\rho * I_x^2 u + G_\rho * I_x I_y v &= -G_\rho * I_x I_t \\
G_\rho * I_y I_x u + G_\rho * I_y^2 v &= -G_\rho * I_y I_t
\end{aligned}
\tag{2.3}
$$

This differential approach was proposed for the computation of small displacements but several extensions (e.g. [13, 5]) have been proposed to overcome this limitation. Mainly, the idea is to embed this computations in a multi-scale approach that would start the by obtaining a rough approximation at a coarse level and successively improve the accuracy of the results and the number of matches by using finer and finer resolutions [5]. The model proposed in [76] assumes an intensity constancy that it does not hold in practice. To overcome this limitation, [93, 47, 13, 53] proposed an extension to gain robustness towards multiplicative and additive changes in illumination. Finally, the local rigidity assumed in equation 2.2 can also be replaced by an affine motion assumption [112] in order to adapt more closely to piecewise flat regions being matched.

Another breed of local matching algorithms perform correlation-based window sweeping to find the best correspondence of a window (in $I_1$) in a second image $I_2$. A complete search of this type is highly time consuming (in the order $O(n^2)$) and can produce several mismatches due to the presence of repetitive patterns or untextured regions. Therefore, the correspondence search is often limited to small neighbourhoods and/or further conditioned (e.g. using epipolar constraints). The matching is commonly performed using the sum of absolute differences (SAD), sum of square differences (SSD) or, more robustly, with normalize cross correlation (NC). Recent window based algorithms reduce the search space by propagating from initial affine invariant features (e.g. [81, 143], see chapter 3). In [54] and [30] a set of features are computed along with their corresponding affine transformations (assuming local planarity). Afterwards, matching information is propagated near the known features

(seeds) using their estimated affine transformation. Once the best match is found their affine transformation is updated. In [30] further strong sidedness and minimum propagations filters are applied which increase the confidence in the propagation while reducing considerably the number of matches. The propagation step can then continue using the next best seed. These, and related methods are detailed in chapter 6.

## 2.3 Global methods

Global methods formulate the matching problem as an optimization of a global energy. Commonly, this energy measures the cost of matching points across two or more images. Such cost can be, for example, the difference between corresponding intensities [50] or the distance between local descriptors [72]. This cost can be computed per pixel, or for a local set of pixels.

The problem of matching points individually is often an ill-posed one. For example, if we want to match a black pixel in an image and there are several black pixels, how do we know who to match to? In global methods, we solve this ambiguity by maintaining coherence and regularity regarding neighboring results. For example, if a pixel $p_1^i$ in $I_1$ matches a pixel $p_2^i$ in $I_2$, another pixel $p_1^j$, neighbor of $p_1^i$ will probably match a pixel neighboring $p_2^i$. Therefore, it is common to add a term that introduces a prior to the formulation so it can be solved in cases where the difference of intensities (data term) is not enough. We then obtain an energy formulation denoted by:

$$E = E_{data} + E_{prior} \tag{2.4}$$

where the first term measures the cost of matching while the second term, usually a measure of smoothness, helps the matching by maintaining local regularity.

### 2.3.1 Variational techniques for global methods

Variational matching methods stem directly from the literature on the optical flow estimation problem. Several of the state of the art methods for dense matching and 3D reconstruction use the variational technique since it produces well behaved results through error equidistribution [120, 6, 111].

Horn and Schunck [50] introduced the variational matching technique for optical flow. This technique aims to reduce the difference in intensities between corresponding pixels in two images by optimizing the displacement vector $[u, v]$ (see equation 2.5) using computations on variations.

Following the model for global matching (equations 2.1 and 2.4) the key idea is to use a linearized version of the squared intensities differences as a data term and the gradient of the displacement field as a regularization term. This formulation is expressed in equation 2.6; where $\alpha$ represents a weight for the smoothness term.

$$\begin{aligned} E(u,v) = \int\int & |I_1(x,y) - I_2(x+u, y+v)|^2 \\ & +\alpha(|\nabla u(x,y)|^2 + |\nabla v(x,y)|^2) dxdy \end{aligned} \tag{2.5}$$

$$\begin{aligned} E(u,v) \approx \int\int & |I_x(x,y)u(x,y) + I_y(x,y)v(x,y) + I_t(x,y)|^2 \\ & +\alpha(|\nabla u(x,y)|^2 + |\nabla v(x,y)|^2) dxdy \end{aligned} \tag{2.6}$$

We will not develop the subject of variational methods here since they are presented and discussed into detail in chapter 4. Furthermore, our own variational matching method for 3D reconstruction will be presented in chapter 5. However, it is important to mention that during the past decades variational methods have gained increasing attention from the research community. Although, the main techniques

in variational methods were developed several decades ago, recent hardware developments have enabled the efficient use of such techniques in the matching problem [6].

### 2.3.2 Discrete techniques for global methods

Another direction that can be taken to solve for global matching methods is to search for the solution in a quantized space. Namely, the problem is solved by assigning corresponding labels to each pixel. Common solving methods following this direction include graph cuts, belief propagation and linear programming. The notion behind them is to approximate the continuous problems with discrete ones, loosing accuracy but gaining in efficiency and robustness.

Several methods in this category merge a set of candidate solutions pre-computed with other method(s) and/or with different setups. For example [103] proposes the fusion of a set of continuous solutions computing minimum graph cuts. [127] proposes to solve a set of binarized continuous subproblems solving for minimum graph cuts using MRF. The motivation in this paper is to overcome the slow convergence speeds that some iterative methods yield by not restricting the flow at each step to local displacements.

Taking advantage of the developments in feature descriptor, the authors of [72] proposed a method that characterizes each pixel using SIFT [74]. The problem is modeled as a minimization of features distances plus a smoothness term. An $L1$ norm is used for the data term and a thresholded $L1$ norm is used for the smoothness term. The problem is then solved using belief propagation. The overall computation is fast but with a loss of precision that manifests itself as a *staircasing* effect in 3D reconstructions.

### 2.3.3 Evaluation data-sets

This work is focused on analyzing and developing matching techniques which work under normal and common conditions. This means that it is important to evaluate the methods in wide-baseline setups, under uncontrolled illumination and using high resolution images. To this end, we have chosen to use the data-sets provided by Stretcha et al. [120] to perform this evaluations because they present such conditions and because, two of the scenes in this data-set present ground truth data obtained using a laser scan. Furthermore, an evaluation methodology is proposed in the same article, which makes it easier to compare to other existing methods that have already published their results.

I some particular cases, we will introduce our own data-sets to present examples of specific situations. For example, in chapter 6 we will discuss different propagation-based methods and we will use many of our own images in varying conditions to exemplify how these methods perform.

In the next chapter we will present the subjects of feature detection and matching. Although these methods provide (mostly) sparse matches which do not allow a full scene representation, they play an important role in the image matching task. Furthermore, they will be used in all the remaining chapters of this thesis.

# Chapter 3

# Feature detection and matching

Stemming from early work on human perception, feature detection and matching has gained great importance in the computer vision community. It is currently applied in tasks such as image retrieval, object recognition, 3D reconstruction, robot localization, image registration, image matching and others. This chapter outlines the body of work on image features, including the tasks of detection, description and matching.

## 3.1 What are local features?

One of the first publications related to features in the field of human perception [4], studies the importance of visual cues, specifically of corners and junctions, for the task of visual recognition. As a computer vision task, first applications of feature detection can be found in robotics, where the author of [86] proposes an obstacle avoidance system for autonomous vehicles.

But, what are local features? Local features can be defined as small portions of images that present distinctive information when compared to that of neighboring areas. They can appear as single points, edges or small blobs and can be used for various applications.

The precise definition of what a feature is varies from application to application. Instead, we can characterize what is looked for when designing and evaluating local features according to five main aspects:

1. Repeatability: The feature detection algorithm should return the same feature locations for the same objects or scenes under varying conditions. In particular, under varying illumination and viewing point.

2. Distinctiveness: A detected feature should be easy to distinguish and therefore easy to match with a corresponding feature.

3. Accuracy: Same features should return similar locations under different conditions.

4. Locality: The feature should be local to reduce the effects of deformations and partial occlusion.

5. Quantity: A large number of features is preferred but prioritizing the distinctiveness. Although this can be argued for the case of object recognition and image retrieval, in the context of this thesis, larger quantities are preferred.

In the next sections we will describe the tasks of detection, description and matching local features.

## 3.2 Feature detection

Feature detection consists in finding image points, lines and regions that present a particular characteristic, commonly defined as a variation or a combination of variations in image properties. These variations can be, for example, an intensity change with a corner-like chape (e.g. [45, 114]) or a constant intensity change along a straight line. Because of the type of information analyzed, feature detectors can be classified in curvature, intensity, color, model and segmentation based detectors.

### 3.2.1 Curvature based feature detectors

In 2D space, curvature is the amount by which a line deviates from being straight. Curvature based detectors aim to find features that present corner-like structures (high curvature). Such feature detectors are known to perform particularly well detecting important points in man-made environments [128].

One of the earliest corner detectors was developed by Moravec [86], where the method searches for the local maximum of minimum intensity changes within a window that shifts along and across the image. A corner will be detected if there is a large intensity variation in every direction around a pixel. This approach was proven to return results with a high level of noise due to the simple use of a sliding rectangular window and due to the effect of rotations not multiple to 45° [86].

#### Harris

The Harris corner detector [45], one of the most used corner detectors, improves on Moravec's work by computing a corner score directly, instead of using shifting windows. The idea is to analyze the (per-pixel) eigenvalues $\lambda_1$ and $\lambda_2$ of the $2 \times 2$ matrix $H_a$ defined in 3.1. If both eigenvalues are found to be large, the location is identified as a corner, if only one is large and the other is close to zero, the location is defined as an edge. In practice, to avoid the computation of eigenvalues, the corner score is approximated by $m = det(A) - \kappa tr^2(A)$, with $\kappa$ a sensitivity parameter. A location will be defined as corner if $m > 0$ and an edge if $m < 0$. This approximation is of less relevance nowadays since eigenvalues can be computed very fast.

$$H_a = \sigma^2 G(s) * \begin{bmatrix} I_x^2(x,y,\sigma) & I_x I_y(x,y,\sigma) \\ I_y(x,y,\sigma) I_x(x,y,\sigma) & I_y^2(x,y,\sigma) \end{bmatrix} \tag{3.1}$$

with $I_x$ and $I_y$ image derivatives obtained using the differentiation scale $\sigma$ following 3.2 and $G(s)$ the gaussian kernel defined in 3.3.

$$
\begin{aligned}
I_x(x,y,\sigma) &= \frac{\partial}{\partial x} G(\sigma) * I(x,y) \\
I_y(x,y,\sigma) &= \frac{\partial}{\partial y} G(\sigma) * I(x,y)
\end{aligned}
\tag{3.2}
$$

$$
G(s) = \frac{1}{2\pi s^2} e^{-\frac{x^2+y^2}{2s^2}} \tag{3.3}
$$

An example of using this detector can be seen in figures 3-1 and 3-2. From top to bottom results are presented for increasing values of $\kappa$.

### Harris-Laplace

The Harris corner detector described before presents high levels of repeatability for translating and rotating cases but fails under varying scaling conditions. The Harris-Laplace feature detector proposed in [81] overcomes this limitation by using an automatic scale selection proposed by T. Lindeberg [67, 70]. The idea behind the scale selection is to find the scale at which the feature detector in use finds a maximum over all sampled scales. Specifically, a gaussian scale space representation is used for the sampling and the Harris detector defined in 3.1 is modified to the Harris-Laplace detector defined in 3.4

$$
M = \sigma^2 G(s) * \begin{bmatrix} L_x^2(x,y,\sigma) & L_x(x,y,\sigma)L_y(x,y,\sigma) \\ L_y(x,y,\sigma)L_x(x,y,\sigma) & L_y^2(x,y,\sigma) \end{bmatrix} \tag{3.4}
$$

where $L(x,y,s) = G(s) * I(x,y)$, with $G(s)$ the circular Gaussian kernel defined in 3.3.

The Harris-Laplace corner detector returns a set of locations $[x,y]$ with a scale factor $\sigma$ which basically describes a circular region which that detected as a feature.

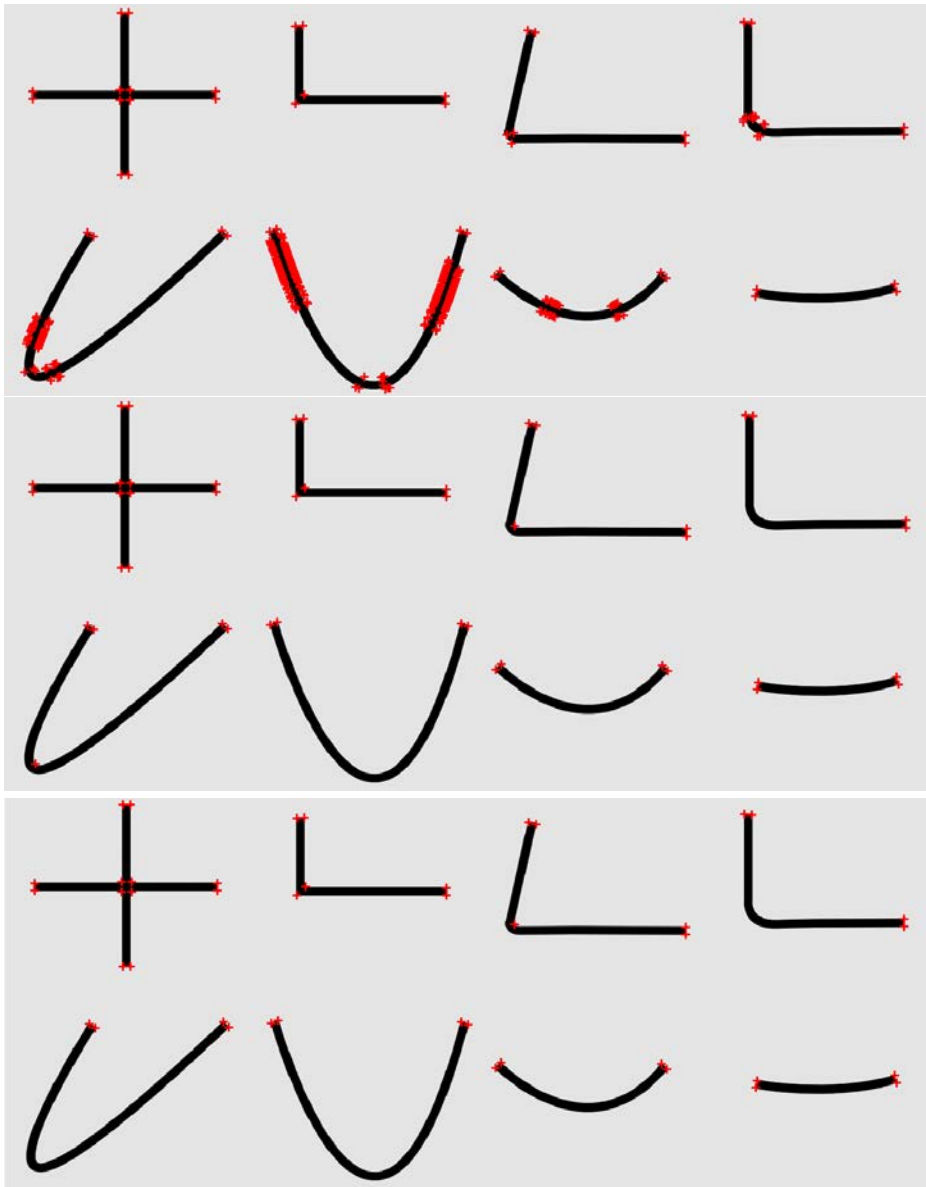


Figure 3-1: Example of the Harris [45] corner detector for increasing sensitivity values. From top to bottom, values taken by $\kappa$ are 0.001, 0.1 and 0.2. This example shows that for low values of $\kappa$ even small curvatures are detected as corners.

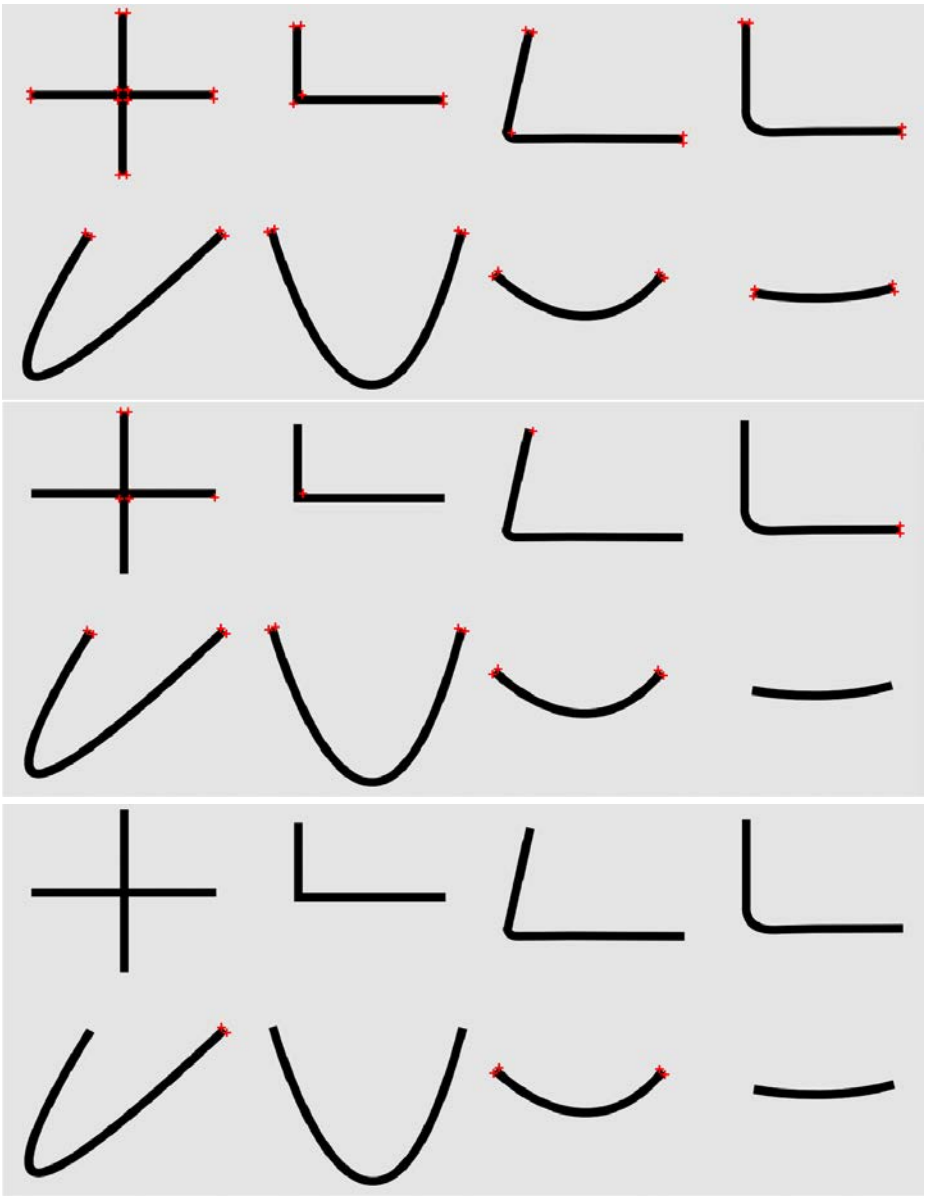


Figure 3-2: Example of the Harris [45] corner detector for increasing sensitivity values. From top to bottom, values taken by $\kappa$ are 0.3, 0.7 and 0.9. This example shows that as we increase the value of $\kappa$ only sharp corners are detected.

#### Harris-Affine

The Harris-Laplace detector makes the original Harris detector work under varying scaling conditions. A further extension to this detector, proposed in [80], aims to estimate the local affine transformation by using the iterative affine region estimation developed by T. Lindeberg [69, 71]. With this method, the detected regions are represented by ellipses that can be transformed to circular regions to obtain affine invariant features (normalization). The Harris-Affine starts by obtaining features using the Harris-Laplace detector previously described and continues with the following iterative steps:

1. Estimate affine shape using normalized second moment matrix
2. Transform ellipsoid region to a circular one (normalization)
3. Re-estimate location of the Harris-Laplace feature
4. If the eigenvalues of the second moment matrix are not equal, go back to step 1

Where the second moment matrix $M_2$ is equal to the matrix $H_a$ defined in 3.1. The normalization step is performed using the root square of the second moment matrix ($M^{1/2}$). Using this detector, the normalized regions of two corresponding points will still relate by an unknown rotation.

### 3.2.2 Intensity based feature detectors

#### Smallest Univalue Segment Assimilating Nucleus: SUSAN

Smith and Brady [114] introduced a corner detector based on a morphological operator. The basic idea is to consider a circular region around a pixel. All the intensity values of the pixels inside this regions are compared to the value of the region's central pixel and classified as Similar or Different. In this way, homogeneous regions will contain almost all of the pixels inside the circular region, as Similar. Near edges, the

Similar pixels should represent around 50%; close to corners they will represent about 25%. Following this observation and giving higher importance to pixels closer to the center, candidate corners are detected where local minimums occur.

**Hessian-Affine**

The Hessian-Affine feature detector works in a very similar way to the Harris-Affine detector and it was proposed by the same authors in [80]. This detector also works in a multi-scale fashion and follows the same steps, but it uses the Hessian matrix to detect feature points at each scale (see equation 3.5). Features are detected when simultaneous local extremas of the trace and determinant of this matrix are found. The determinant of $H_e$ reaches a maximum for regions of the images whit blob-like structures.

$$H_e = \sigma^2 G(s) * \begin{bmatrix} L_{xx}(x,y,\sigma) & L_{xy}(x,y,\sigma) \\ L_{xy}(x,y,\sigma) & L_{yy}(x,y,\sigma) \end{bmatrix} \tag{3.5}$$

with $L_{xx}$, $L_{xy}$, $L_{yy}$ second partial derivatives obtained in a similarly to 3.2.

**Difference of Gaussians: DoG**

Feature detectors based on DoG detect space-scale extremals of the difference of two gaussian functions (see 3.3) convolved with the target image $I$. This difference can efficiently be computed from the difference of two images ($L_a$ and $L_b$) obtained by convolving image $I$ with two gaussian kernels that differ by a small scaling factor (see equation 3.6). This technique was proposed by Lowe [75] based on studies on Gaussian kernels for scale-space computations(e.g. [58, 67, 68]).

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) = L(x,y,k\sigma) - L(x,y,\sigma) \tag{3.6}$$

with $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$ and $G(x, y, \sigma)$ the circular Gaussian kernel defined in 3.3.

### Scale Invariant Feature Transform: SIFT

The SIFT algorithm [74] starts by detecting extremas using the multi-scale DoG method previously described. It proceeds by removing poor features and refining the interesting ones. This filtering and sub-pixel refinement works by fitting a second-order Taylor expansion of the 3D quadratic surface (in $x$,$y$ and $\sigma$). Namely, equation 3.6 is approximated by 3.7 using $z_0 = [x_0, y_0, \sigma_0]^T$ and $z = \lambda[x, y, \sigma]$; and the extremum is located by setting the derivative of 3.7 with respect to z equal to zero.

$$D(z + z_0) = D(z_0) + \left(\frac{\partial D}{\partial z}|_(z_0)\right)^T + \frac{1}{2} z^T \left(\frac{\partial^2 D}{\partial z^2}|_(z_0)\right) z \tag{3.7}$$

This process is repeated since areas under consideration change with every movement. Points that do not converge quickly are filtered out.

### Affine SIFT: ASIFT

ASIFT [87, 143] aims to to add fully affine invariance to the, already rotation and scale invariant, SIFT algorithm. It assumes that when two pictures of a solid piecewise smooth object are taken by cameras in different positions, the resulting images follow apparent smooth deformations. These smooth deformations can be closely approximated by local affine transformations. Similar assumptions are made in the cases of Harris-Affine and Hessian-Affine, but ASIFT proceeds in a different manner. Instead of working simply with image information to gain invariance, ASIFT simulates a set of possible image views obtained by exploring the two orientation parameters of the camera. This is, in nature, the same procedure taken by most feature detectors to behave in a scale covariant manner: simulate variations. In the case of ASIFT, the

local image deformations are modeled by: $I_1(x, y) = I_2(ax + by + e, cx + dy + f)$ and the local transformation matrix $A$ can be defined and decomposed as:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \lambda \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} t & o \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3.8}$$

with $\theta$ the longitude angle between the optical axis and a fixed vertical plane, $\arccos(1/t) = \phi$ the latitude angle between the optical axis and the image plane normal, $t > 1$ the tilt, $\phi$ the camera rotation angle around the optical axis, $\lambda$ a zoom parameter. See figure 3-3 for a graphic representation of the model. ASIFT, densely simulates different transformations by varying the zoom, latitude and longitude and computes a SIFT descriptor for every simulated view and pairwise matches are obtained.

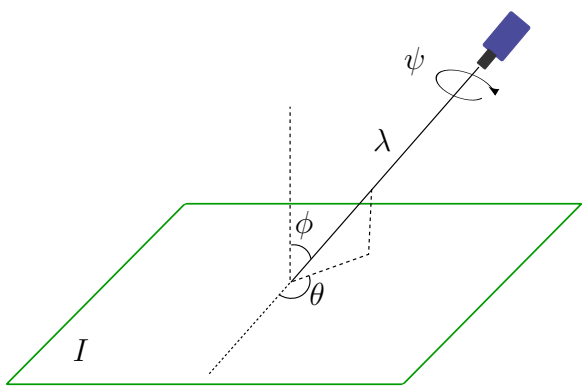



Figure 3-3: Geometric representation of equation 3.8.

### Speed Up Robust Features: SURF

The Speed Up Robust Features (SURF) proposed in [10, 9] are an scale-invariant feature detector based on the Hessian matrix. The main idea is to approximate the

Hessian matrix using box-type filters employing integral images [134]. The determinant of the Hessian is then used to determine the location of the feature as well as the characteristic scale. The approximation consists in replacing the Gaussian second-order partial derivative by the response to approximated filters which can be computed extremely fast. An illustration of the filters used is shown in figure 3-4. As with other Hessian-based feature detectors, the SURF feature detects blob-like structures in the images.

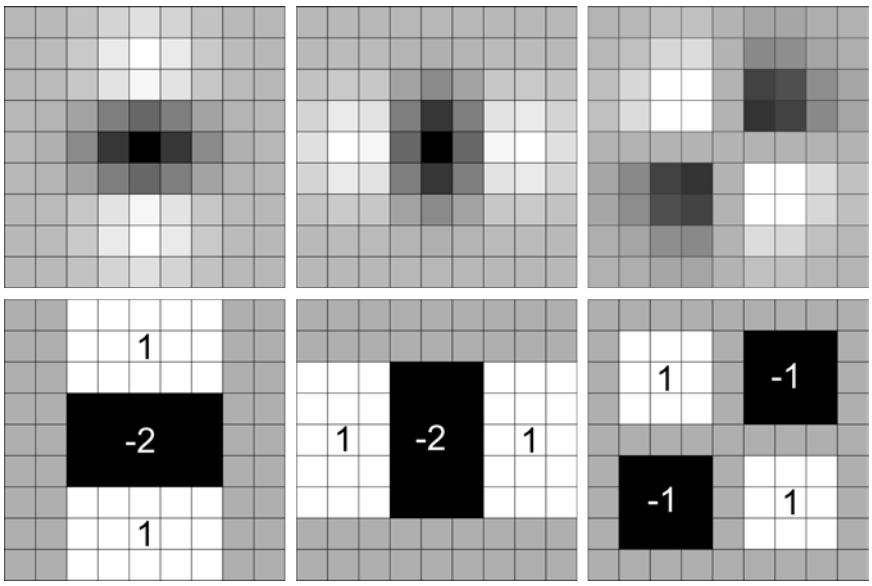



Figure 3-4: SURF filter approximations. Top row, discretized Gaussian second order partial derivatives ($G_{yy}$,$G_{xx}$ and $G_{xy}$). Bottom row, corresponding SURF box filter approximations.

### Intensity-based regions: IBR

An affine invariant feature detector based on image-intensities was proposed by Tuytelaars and Van Gool [130, 129]. Their method starts by obtaining multi-scale intensity extremas and continues by exploring the image around them. This exploration is

done by evaluation the function 3.9 over rays emanating from the extremas (see figure 3-5). A maximum is reached over each ray when sudden changes occur and all the maximums are linked to the center (feature) point to create a (feature) region. Finally, an ellipse is fitted to this region to represent it by a $2 \times 2$ affine transformation (see figure 3-5).

$$f(t) = \frac{|I(t) - I_0|}{max\left(\frac{\int_0^t |I(t)-I_0|dt}{t}, \epsilon\right)} \tag{3.9}$$

with $\epsilon$ a small number to avoid division by zero.

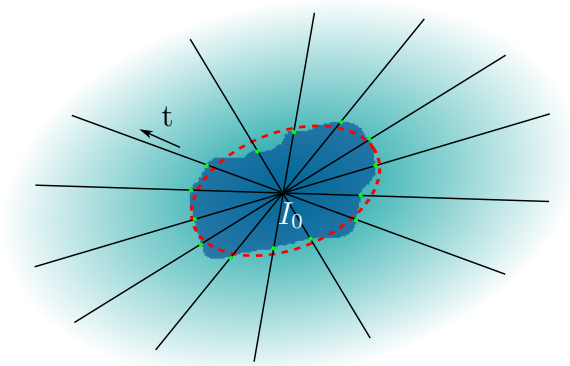



Figure 3-5: Intensity-based region detection. Rays emanating from $I_0$ find a maximum at the boundaries of the region (green dots). The dotted (red) ellipsoid represents the affine transformation used for later normalization.

### 3.2.3 Segmentation based features

#### Maximally stable extremal regions: MSER

The MSER proposed by Matas et al. [79] aims to detect blobs in images. An MSER region is a set of connected elements that have either higher or lower intensity that all the pixels in the outer boundary of the region. To find this regions, all the pixels in an

image are first sorted by intensity and merged using the union-find algorithm [110]. The intensity value at the location of the minimum rate of change of intensity inside a region is used to threshold the image and produce the final (connected) set of MSER. Evidently, the MSER region detector works best in structured images while it was found to have problems coping with blurred edges [83].

### 3.2.4 Model based feature detectors

Model based descriptors aim to provide a formal representation of corner points in an image. One example of such representation is presented in [41], where the author models a corner as a blurred wedge, parameterized by its angle, amplitude and blur. Features were detected by fitting this model into local images. Other models proposed include junctions [102] that are also detected by fitting a parametric model that comprises several homogeneous regions with blurred junctions. Other methods, for example, aim to characterize the scale response of common feature detectors [27]. In general, this breed of methods suffer from an important drawback: high complexity, which makes difficult the design of rich feature detectors, and makes for high running times.

### 3.2.5 Color based feature detectors

In the previously outlined methods, feature detection was performed based only on image intensities, either by differentiation or by directly analyzing the intensities. Color based detectors build-up on the previous ideas, combining them with color distinctiveness. Examples include the CSIFT method [1], which presents an extension to the SIFT detector and descriptor based on a color invariance model [38].

## 3.3 Feature descriptors and matching

Ideally, feature descriptors should be distinctive and invariant to geometric and photometric transformations. The simplest descriptor of a region around a feature point or inside a feature region is an array of pixels. Arrays of intensity valued pixels can be used to perform matchings across images using some correlation score.

### 3.3.1 Feature descriptors based on distribution

#### SIFT

The SIFT descriptor proposed by D. Lowe [74] basically consists of a normalized vector with 128 values. This vector represents a histogram of orientations of image gradients, normalized by the magnitude of the same gradients. A grid of size $4 \times 4$ is constructed around a detected feature point and inside each grid, the image gradients are classified in eight different bins ( $4 \times 4 \times 8 = 128$), see figure 3-6. The contribution of each gradient to the histogram will depend on its magnitude and a weight given by a Gaussian function around the feature point. The overall extension of the $4 \times 4$ grid will depend on the scale at which the feature was detected.

Since the obtained descriptor is of only 128 (constant) dimensions, in [74] is proposed to use nearest neighbor to perform feature matching. This will return matches for all the features detected, including those that do not actually have a correspondence. Therefore, matches are only preserved if the distance to the second nearest neighbor is significantly larger.

#### Gradient location-orientation histogram: GLOH

The GLOH [82] descriptor aims to extend SIFT by performing principal component analysis (PCA) [52] to reduce the dimensions of the feature vectors. It works by computing SIFT descriptors for a log-polar location grid with 8 bins in the angular direction and 3 bins in the radial direction, which results in 17 location bins (central

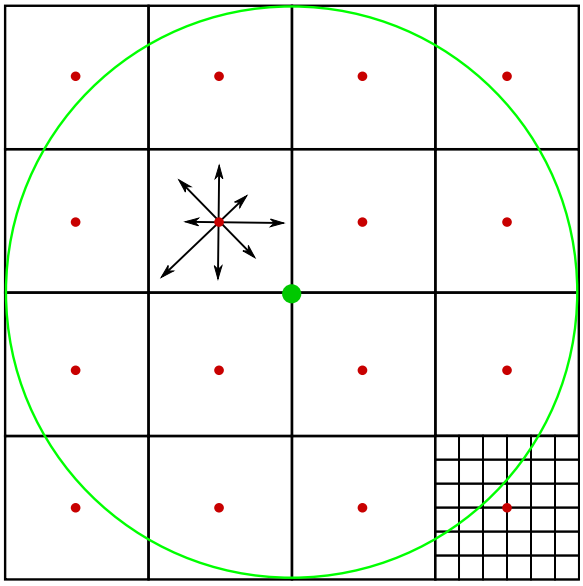


Figure 3-6: Illustration of the grid used to compute the SIFT descriptor. The green dot represents the location of the feature detected. The cell with the arrows represents a histogram with the eight bins. In the lower right cell is exemplified a case for $6 \times 6$ pixels.

bin is not divided in different angular directions). Furthermore, gradient orientations are divided in 16 bins, giving a total of 272 bins. This number of dimensions is reduced using PCA by keeping the 128 (or 64) largest eigenvectors.

### 3.3.2 Feature descriptors based on filters

Filter based descriptors are based on the response of three main filters: Gabor filters [26], Steerable filters [32] and complex filters [108]. For example, from a cognitive point of view [78], Gabor filters based on image decomposition are suggested to be relevant to human image understanding. The Gabor filters consist of a set of local bandpass functions, which react to orientation and frequency. In the context of complex filters, [108] proposes a filter that can have 16 different responses, similar to the derivatives of a gaussian function. The filter used is formulated as:

$K_{mn} = (x + iy)^m (x - iy)^n G(x, y)$, with $G(x, y)$ is the Gaussian function, $m + n \leq 6$ and $m \geq n$.

### 3.3.3 LDAHash

The SIFT descriptor described previously describes features using a vectors of 128 dimension. Matching feature points using such descriptors works reasonably for a few hundreds of points or less, but it becomes a problem for thousands or millions. The LDAHash feature matching method proposed by Strecha et al. [20] minimizes that problem by mapping the descriptors into the Hamming space. Once in this space, the Hamming metric is used to compare the resulting descriptors.

The Hamming space [43] is the set of $2^N$ binary strings of length $N$. The Hamming distance between two strings is the number of positions at which they differ.

The LDAHash method first finds an affine mapping function that minimizes in-class covariance and maximizes across-class covariance. Then, it computes a threshold to binarize the mapped descriptors.

### 3.3.4 Feature descriptors based on spin images

Spin images [51] represent the distribution of intensities around a central point in a normalized histogram. Each histogram is obtained for 5 to 10 rings around the central point. They are usually used in conjunction with normalized regions from Harris-Affine or Hessian-Affine feature detectors.

### 3.3.5 Feature descriptors based on color

Color based image feature descriptors aim to use information from color images instead of simply grayscale versions. For example, [133] present distribution-based color descriptors. Normalized RGB, hue, opponent angle and spherical angle are used to

create histograms. In general, combinations of color and shape-based descriptors can provide richer information to perform matching and recognition [133].

## 3.4 3D reconstruction using image features

Sparse sets of features can provide important information for tasks such as object obstacle avoidance and scene calibration. They are also used in the process of automatic scene calibration (e.g. [73, 10, 123]). However, they are often not envisioned for full scene reconstruction. Dense feature descriptors and matching methods (e.g. [124, 20, 72]) propose a way to overcome this limitation by computing per-pixel feature descriptors that are matched across images.

### 3.4.1 DAISY

Tola et al. [124] proposed a local descriptor that can be efficiently computed and matched in a dense manner. Motivated by SIFT, the DAISY algorithm computes descriptors based on orientations. It starts by computing quantized orientation maps for several predefined directions. Each map represents a direction $\theta$ and it stores the gradient norms for the locations where gradients are positive in the direction of $\theta$. Each map is convolved with Gaussian kernels of different scales. Once these maps are obtained, they can be efficiently used to obtain histograms of orientations for different area sizes, with different number of sub-areas and and different numbers of bins. An important speedup over the computation of SIFT descriptors is that histograms of neighboring points are mostly similar and therefore can be reused with minimum amount of work. The matching process is formulated as an expectation maximization algorithm following [14] and [59].

### 3.4.2 SIFT Flow

The SIFT Flow method [72] works by matching pixel-wise SIFT [74] descriptors across two images. The problem is formulated as an energy minimization that optimizes for the integer displacement vectors $u(x, y)$ and $v(x, y)$. Initially, the sift-images $S_1$ and $S_2$ are computed by obtaining the per-pixel SIFT descriptors over two images $I_1$ and $I_2$. Then, the energy function 3.10 is minimized using belief propagation [89] in a coarse to fine scheme.

$$E(u(x,y),v(x,y)) = \sum_x \sum_y min(\|S_1(x,y) - S_2(x+u(x,y), y+v(x,y))\|_1, \epsilon) + \sum_x \sum_y \sum_m \sum_n min(\beta|u(x,y) - u(m,n)|) + min(\beta|v(x,y) - v(m,n)|) + \sum_x \sum_y \alpha(|u(x,y)| + |v(x,y)|, \gamma) \tag{3.10}$$

with $[m, n]$ neighboring positions of $[x, y]$ in a four connected scheme, $\alpha$ and $\beta$ weighting function and, $\epsilon$ and $\gamma$ thresholding values. The first term in the function is used to minimize distance between descriptors, the second term works as a regularizer and the last term constraints the displacement vectors to be as small as possible.

## 3.5 Experiments

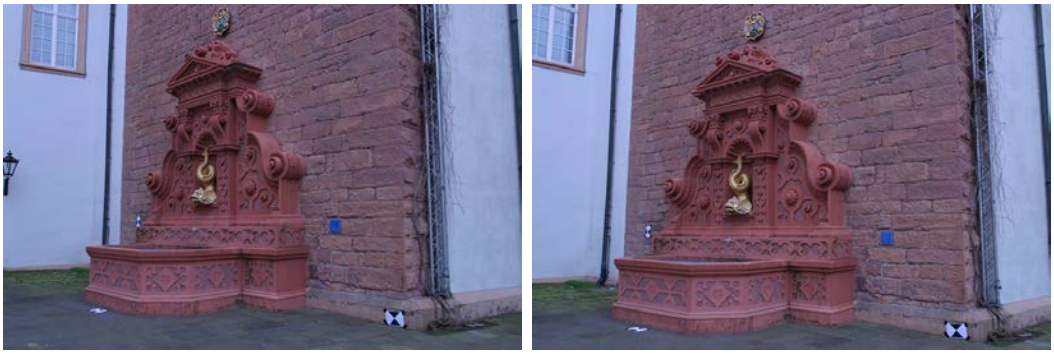
In this section we illustrate the performance of several feature detecting and matching techniques that include Harris-Affine, Hessian-Affine, SIFT, SURF, ASIFT and LDAHASH. Benchmarks on the Fountain-P11 scene [120] are shown in figures 3-7 to 3-9. The pairs of views evaluated in this scene are: $0 - 1$, $0 - 2$, $0 - 4$, which approximately correspond to baselines of angles 10.5, 21, and 43 degrees. The charts

display the true and false matches per method. For the current tests, a matching is considered correct if the distance to the ground truth is less than one pixel.

We present similar evaluations performed on our painting dataset in figures 3-10 to 3-12. The three pairs present three important matching scenarios: i) scaling, ii) rotation and iii) slanted view that includes rotation and scaling. The two scenes are different in nature, the fountain scene presents highly textured images with small structures and several orientation changes. The second dataset presents more structured images that include only one surface which can be estimated by a plane. Notice the general superiority of the LDAHASH feature matching and detection technique, which reliably returns a high number of positive matches accompanied of a low number of false matches. ASIFT, on the other hand, presents a high number of false matches, which in many times (4 out of 6 experiments) surpasses the number of positive matches. Further actions can be taken to filter out these false matches (e.g. RANSAC) but there is a high probability of accepting a false match.

## 3.6 Conclusion

In this chapter we have presented several feature detectors and several feature matching techniques. Although it certainly does not cover all the material related to features, it provides the main concepts that will be referred and used in the next chapters. In the next chapter, we will show how sparse feature matchings can be introduced as anchor points in order to guide the variational matching. In chapter 6, Harris corner detectors will be used to automatically start a propagation based matching. Later in the same chapter we will show, how affine invariant feature detectors (e.g. Harris affine/Hessia affine), provide important additional information that enables better performing match propagations.



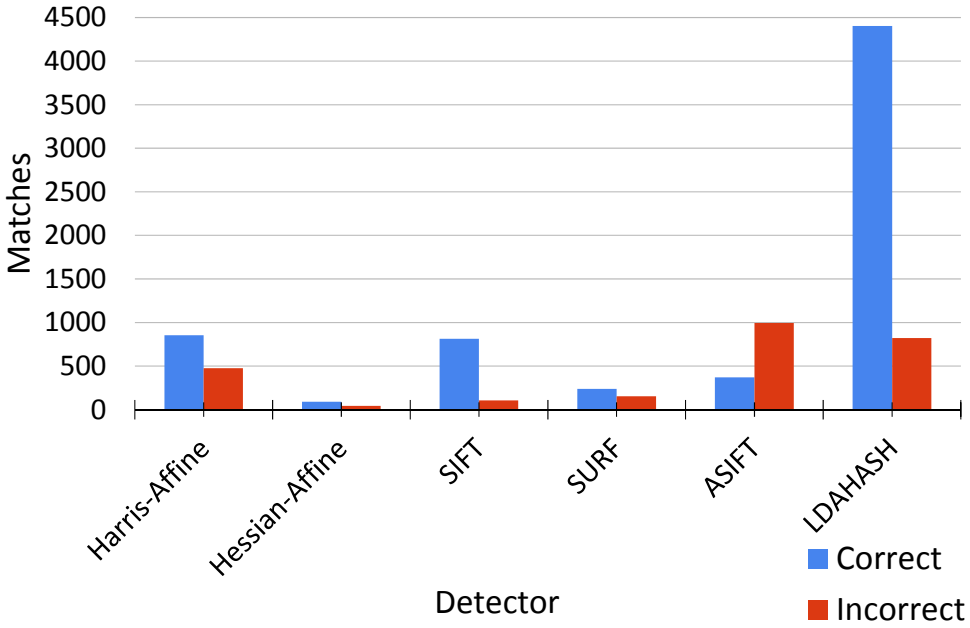
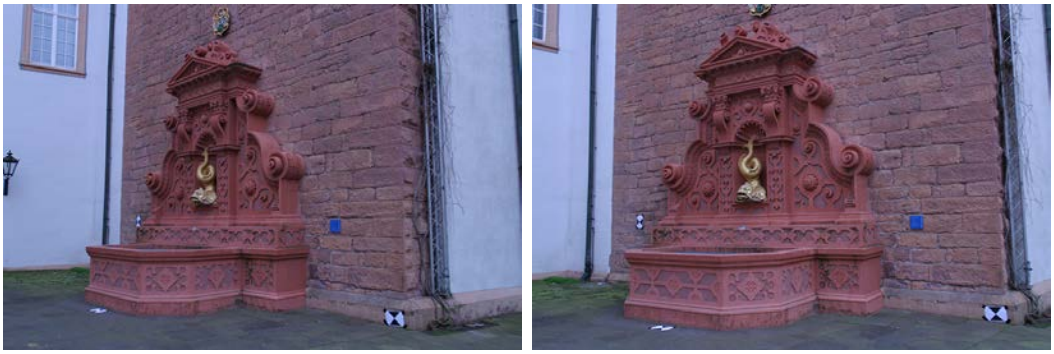



Figure 3-7: Fountain scene, views $0-1$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red



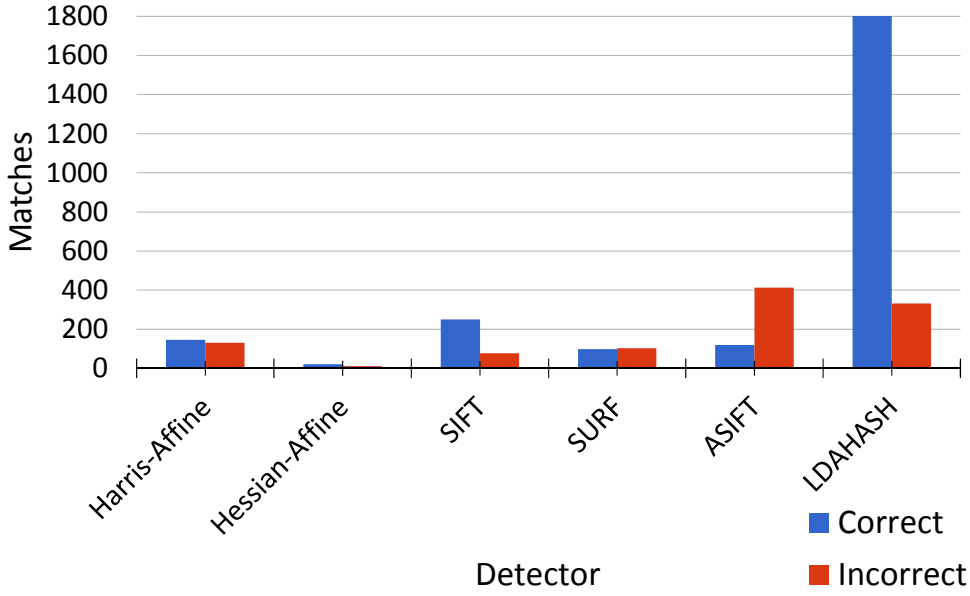
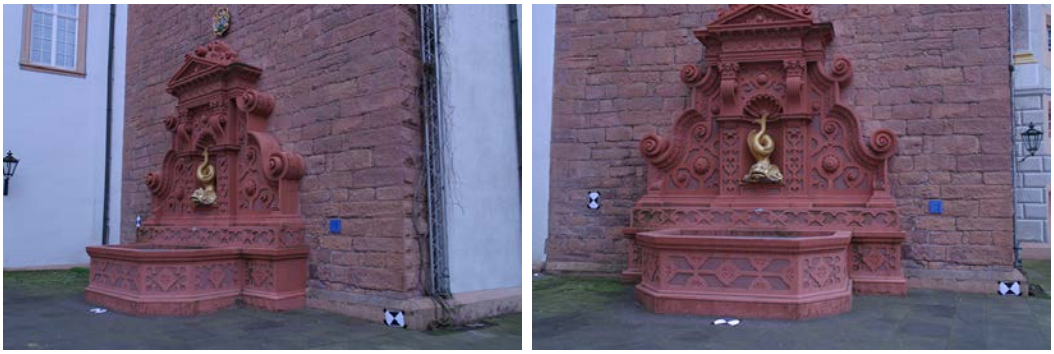



Figure 3-8: Fountain scene, views $0-2$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red



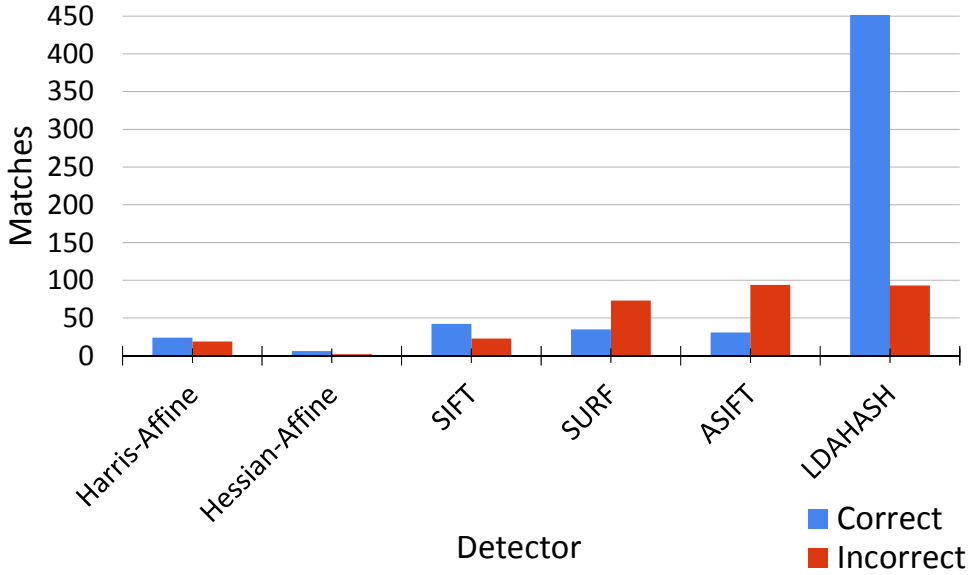



Figure 3-9: Fountain scene, wide-baseline pair of views $0-4$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red

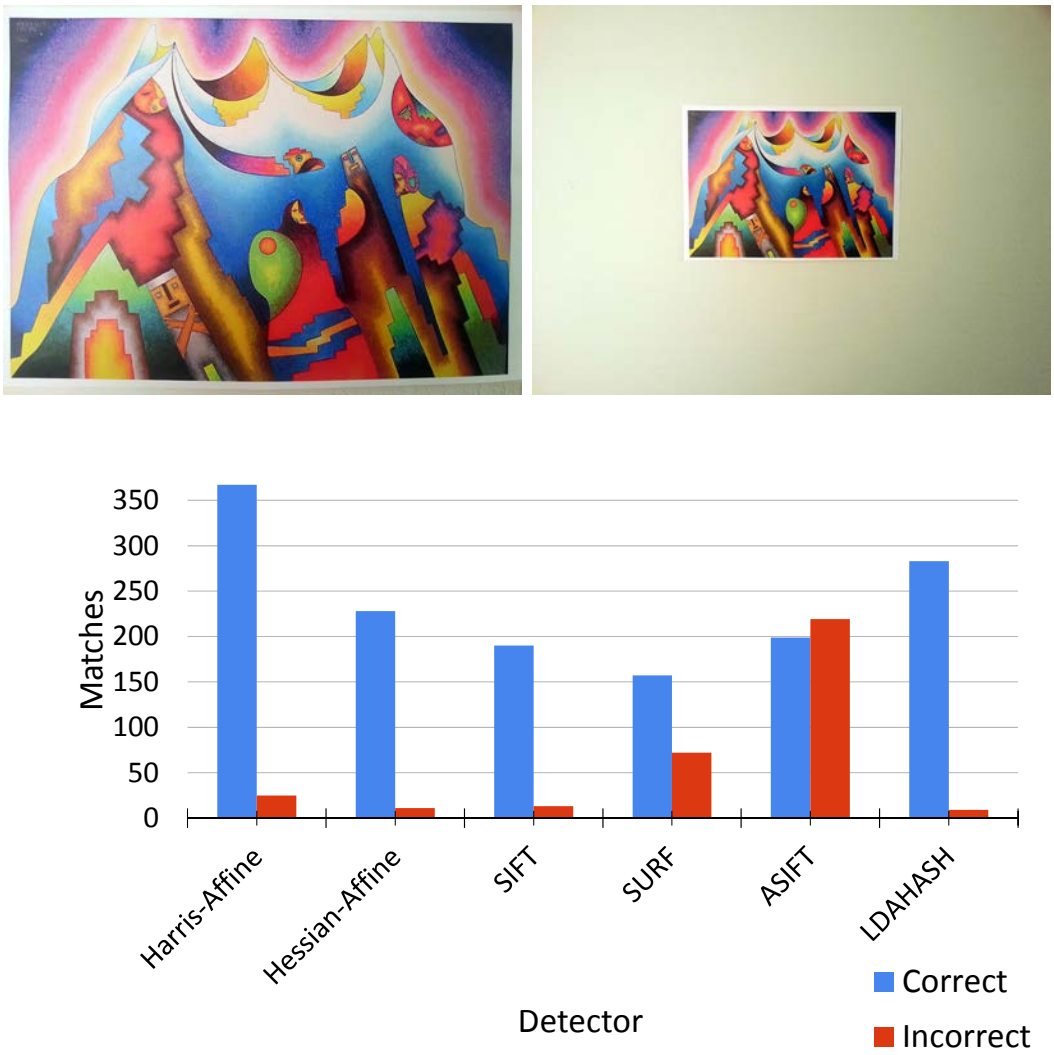



Figure 3-10: Zoom out example for the painting dataset, views $0 - 1$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red

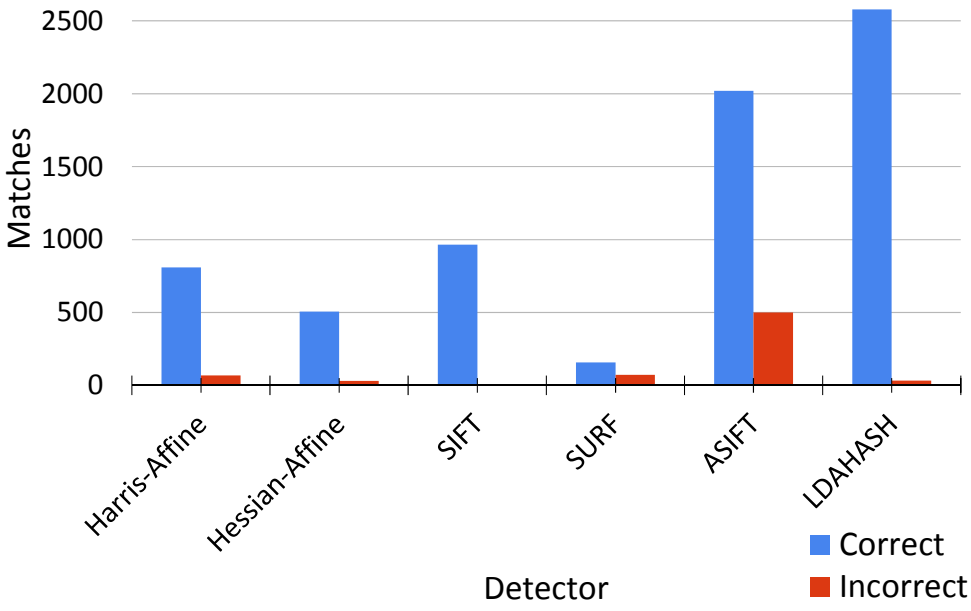



Figure 3-11: Rotation example for the painting dataset, views $0-2$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red

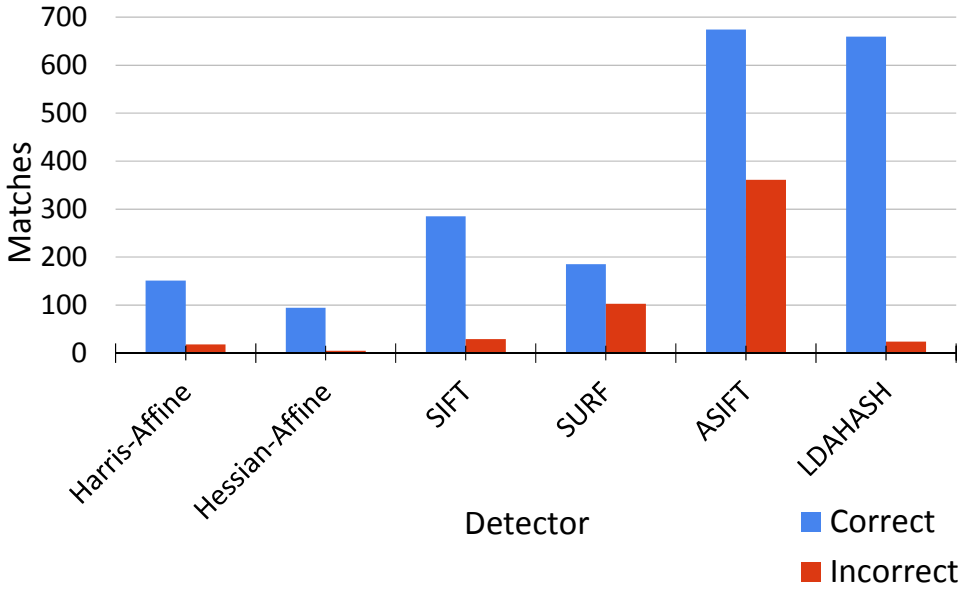



Figure 3-12: Slanting, rotation and scaling for the painting dataset, views $0-3$ (top). In the chart (bottom), correct matches are illustrated in blue, wrong matches in red

# Chapter 4

# Variational Image Matching for 3D reconstruction

This chapter reviews variational methods as a way to find image matches. The techniques described herein are mainly targeted to the estimation of matches across two images which belong to a set of multiple images of the same scene. Following the basic concepts of variational matching, several methods have been developed and evolved to a great extent. We start with the seed work of Horn and Schunck proposed in the context of optical flow [50], which works well for simple problems. We continue reviewing advances that build on this formulation and improve on its speed, robustness and its ability to capture sharp discontinuities,

We present basic notions of variational matching along with two extensions that use further constraints to improve resulting 3D reconstructions. These two extensions take advantage of scene geometry by using pre-computed sparse feature matching and epipolar constraints. First, the use of a sparse set of features can bring improvements to the results in certain conditions such as large displacements of small objects. These features are filtered using knowledge about the scene's geometry, namely, the projection matrices or at least the fundamental matrix. Second, epipolar information as a

constraint can also guide the variational matching process without the need for image rectification and subsequent interpolation errors.

## 4.1 Classical formulation

The work of Horn and Schunck [50] introduced a matching method along with a variational technique to solve its application (see sub-section 2.3.1). The key idea is to formulate the problem as an energy minimization that includes two terms, one that quantifies intensities' similarities and another that quantifies local smoothness. Namely, the squared intensities differences is used as a data term and the squared sum of the gradients of the displacement field is used as a regularization term. This formulation is expressed in equation 4.1; where $\alpha$ represents a weight for the smoothness term.

$$E(u,v) = \int\int |I_1(x,y) - I_2(x+u, y+v)|^2 + \alpha(|\nabla u(x,y)|^2 + |\nabla v(x,y)|^2)dxdy \tag{4.1}$$

The data term in this equation is non-linear in $u$ and $v$ but can be linearly approximated by a first order Taylor expansion:

$$E(u,v) \approx \int\int |I_x u + I_y v + I_t|^2 + \alpha(|\nabla u|^2 + |\nabla v|^2)dxdy \tag{4.2}$$

with $I_x = (I_{2x} + I_{1x})/2$ , $I_y = (I_{2y} + I_{1y})/2$ , $I_t = I_2 - I_1$ and subscripts $x$ and $y$ denoting partial derivatives.

In the calculus of variations, Euler-Lagrange characterizes the optimization process for the energy above, as finding $u$ and $v$ such that:

$$\begin{aligned} (I_x u + I_y v + I_t) I_x - \alpha \triangle u = 0 \\ (I_x u + I_y v + I_t) I_y - \alpha \triangle v = 0 \end{aligned} \tag{4.3}$$

with reflecting boundary conditions.

## 4.2 Data term

The data term in the variational matching formulation represents the driving force during the optimization process. Horn and Schunck proposed a linearized version of the squared difference in image intensities (see equation 4.2). This approximation is practical and the quadratic function can be easily optimized. However, it presents important drawbacks. First, the brightness constancy assumption used in this model, most often does not hold since it assumes that the illumination is constant, that the surface reflectance is lambertian and that the acquisition method is perfect (noise free). Furthermore, the squared nature of the data term penalizer is too sensitive to noise, outliers and occlusions and does not closely represent the problem that we are trying to solve. In more detail, quadratic penalizers give a strong influence to points that are far from the assumptions. Since each difference is squared, the solution is adapted to fit the points with larger errors, simply not accepting the existence of outliers [16].

### 4.2.1 Data term penalizers

Several ways to improve on the square penalizer have been studied. In general, we seek to accept outliers by using a robust penalizer [121, 12, 18]. For example, a penalizer of the form $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ is proposed in [18]. In this function $\epsilon$ represents a

very small value that makes the function continuously differentiable in first degree. This robust $\Psi(s^2)$ function will be used in all our following methods unless stated otherwise. The same follows for all the evaluations and experiments that make use of variational matching.

In the left chart of figure 4-1 we illustrate the shape of the penalizing functions. We can observe that the robust function presents a sharp discontinuity. In the same figure, to the right, we can observe the effect of the penalizer in the equations being solved. This effect is given by the first derivative of the function. We can perceive that the robust penalizer varies sharply and saturates at both ends, effectively allowing us to capture sharp discontinuities. On the other hand, the square penalizer varies linearly and continues increasing at both ends.

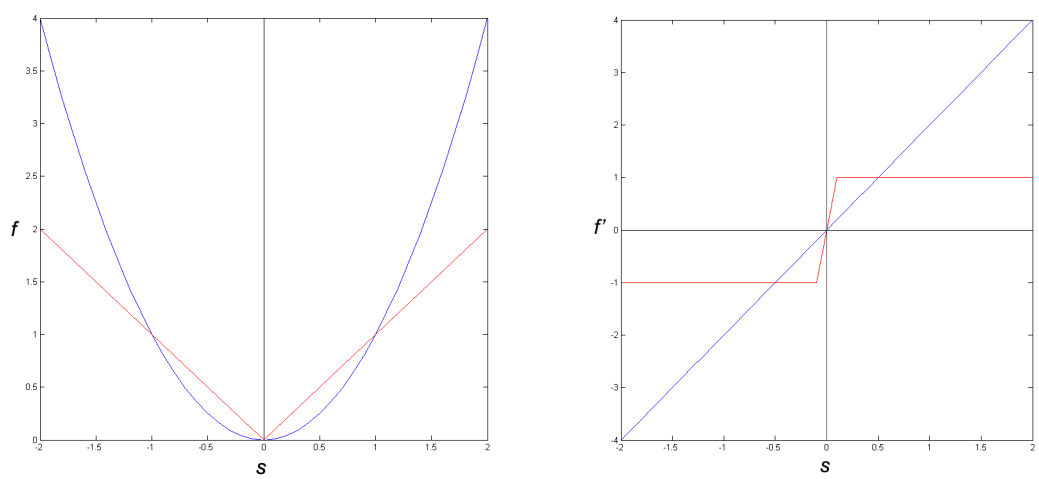



Figure 4-1: To the left, shape of the penalizers $f(s^2) = s^2$ (in blue) and $f(s^2) = \sqrt{s^2 + \epsilon^2}$ (in red) for $s = [-2, 2]$. To the right, influence of the square penalizer given by $f'(s^2) = s$ (in blue) and influence of the robust penalizer given by $f'(s^2) = s/\sqrt{s^2 + \epsilon^2}$ (in red)

### 4.2.2 What to measure

The intensities constancy assumption can be complemented or even replaced by other measures in order to provide methods more robust to illumination changes and other optical variations. In the next paragraphs we discuss the use of gradient constancy, structure-texture decomposition, affine covariant intensity model, invariant features and color as a way to improve the data term.

#### Gradient constancy

One way to reduce the effect of illumination changes is to use the difference between image gradients as the primary energy term [18]. This approach is expressed in equation 4.4, where only the data term is presented using the robust penalizer just explained. In this equation $I_x$ and $I_y$ represent partial derivatives in x and y. In particular, this approach gains robustness against additive illumination changes.

$$\begin{aligned} E(u,v) = \int\int \Psi(|I_1(x,y) - I_2(x+u,y+v)|^2) + \\ \alpha(\Psi(|I_{1x}(x,y) - I_{2x}(x+u,y+v)|^2) + \Psi(|I_{1y}(x,y) - I_{2y}(x+u,y+v)|^2)) \end{aligned} \tag{4.4}$$

#### Structure texture decomposition

Another approach, presented in [104, 136], performs a Structure-Texture decomposition to pre-process the input images and reduce brightness changes. Each pre-processed input image is decomposed in structure and texture and, ideally, these two terms can be re-combined to create an (illumination) invariant version of the image. For example the structure-texture combination can be in the proportion $1:20$ [121]. Following [137], the structure-texture is obtained by decomposing $I = I_s + I_t$, and minimizing equation 4.5.

$$E(I_s) = \int_\omega \left( |\nabla I_s| + \frac{1}{2\theta}(I_s - I)^2 \right) \tag{4.5}$$

where $\theta$ is a small constant.

This structure-texture pre-processing of the images improves results obtained with the classical Horn and Schunck method. However, similar results can be obtained with the simple gradient constancy assumption previously presented [121].

### Affine covariant intensity model

We can also gain robustness to illumination changes by considering bias and gain parameters, as suggested in [76, 106]. Specifically, the data term can be formulated as:

$$\begin{aligned} E(u,v) = \int\int \Psi(|g(x,y)I_1(x,y) + b(x,y) - I_2(x+u, y+v)|^2) \\ +\phi\Psi(|\nabla b(x,y)|^2) + \mu\Psi(|\nabla g(x,y)|^2)dxdy \end{aligned} \tag{4.6}$$

where $b(x,y)$ represents a bias parameter and $g(x,y)$ represents a gain (offset) parameter. In the second line of the function, a regularization term is used to make this parameters vary smoothly across the image, where $\phi$ and $\mu$ are weights. $b$ and $g$ can be initialized as 1 and 0 everywhere and, they are continuously adjusted inside an iterative solving procedure, which alternates between solving for $[u, v]$ and solving for $[b, g]$.

### Invariant features

Descriptors and non-parametric transformations as wavelets [139, 92], census transform [116] and HOG [19] can also be used to gain even greater robustness and speed

while reducing accuracy. For example, [19] proposes to use a local descriptor matching to augment a data term similar to 4.4. The used feature descriptors are Histograms of Oriented Gradients (HOG) [25] (see chapter 3) combined with the mean color and the feature matching is performed in advance for every point. Such data term can formulated as:

$$\begin{aligned} E(u,v) = \int\int \Psi(|I_1(x,y) - I_2(x+u,y+v)|^2) + \\ \alpha(\Psi(|I_{1x}(x,y) - I_{2x}(x+u,y+v)|^2) + \Psi(|I_{1y}(x,y) - I_{2y}(x+u,y+v)|^2)) + \\ \kappa\Psi(|u(x,y) - u'(x,y)|^2 + |v(x,y) - v'(x,y)|^2) \end{aligned} \tag{4.7}$$

where $[u', v']$ represent the pre-computed displacement vectors. The term is added to 4.4 to stay close to what is proposed in [19].

### Color based terms

Nowadays, grayscale images are mostly used for artistic reasons while mainstream pictures contain color representation. Typical digital color representation consists in combining three base channels: red, green and blue (RGB). Therefore, a direct extension to the classical variational optical flow method is to simply use one data term for every channel (e.g. [90, 8]).

More involved color-based optical flow methods have been proposed based on the HSV color space and treating the three resulting bands (hue, saturation and value) differently by using independent penalizers (e.g. [147]). The HSV color space presents higher invariance to photometric changes. The Hue channel is invariant to multiplicative illumination changes (e.g. shadows, shading, highlights and specularities). The saturation channel is only invariant to shadow and shading while the value channel is not invariant to any of these changes since it encodes the actual brightness. The HSV color space represents RGB in a cone setup. Another way to represent RGB is to use a

spherical transform as it is done, for example in [84]. In their method, three channels are obtained using the spherical transformation, which represent magnitude, longitude and latitude. The two angles (magnitude and latitude) can be used to compute the optical flow since they are invariant with respect to shadow and shading.

## 4.3 Regularization

The use of a quadratic regularization term in equation 4.2 is very convenient since it leads to a linear optimization problem. However, it does not fairly represent the model that we are trying to formulate since it gives too much influence to outliers and does not allow to closely follow discontinuities [16].

To overcome this problem many authors (e.g. [18, 12]) have proposed the use of functions from robust statistics, similarly to what is done with the data term in subsection 4.2.1. The commonly used function is also of the form $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$. This function approaches an L1 norm (see subsection 4.2.1). Other functions can be used, for example the Lorentzian [12] function, that has the form $\Psi(s^2) = \log\left(1 + \frac{1}{2}(s/\rho)^2\right)$, where $\rho$ is an scale parameter.

Using a robust penalizer, the regularization term in equation 4.2 is replaced by $\alpha\Psi(|\nabla u|^2 + |\nabla v|^2)$, and the complete Euler-Lagrange equations can be denoted as:

$$\begin{aligned} &\Psi'((I_x u + I_y v + I_t)^2)(I_x u + I_y v + I_t)I_x - \alpha div(\Psi'(|\nabla u|^2 + |\nabla v|^2))\nabla u) = 0 \\ &\Psi'((I_x u + I_y v + I_t)^2)(I_x u + I_y v + I_t)I_y - \alpha div(\Psi'(|\nabla u|^2 + |\nabla v|^2))\nabla v) = 0 \end{aligned} \tag{4.8}$$

Note that the term $\Psi'(|\nabla u|^2 + |\nabla v|^2))$, which multiplies the gradient of $u$, can be interpreted as a diffusion term and, as shown in [99], it does not need to be mathematically exact to quickly converge to the best results. What remains important is that through this term we can introduce a strong weight to enforce the smoothness of results or a low weight to respect sharp features. Moreover, a matrix can be used

to obtain anisotropy in the smoothness process, smoothing in one direction while allowing sharp discontinuities in the other.

In the previous case the smoothness weight is determined by information obtained from the displacement vector fields, where a high variation in the displacement will give a low importance weight and a low variation will return a high importance weight. Following this basic principle, other cues can be introduced through this term for the purpose of a having a more sensitive regularizer or even for guiding the matching process. For example, the method proposed in [100] uses a regularization term weighted by motion segmentation that is obtained from previous computations. Besides information related to the motion field, image information has been used in this term. For example, [138] uses image edges to reduce the effect of the regularization term across them, effectively formulating the regularizer as:

$$\Psi((\nabla u)^T D \nabla u) \tag{4.9}$$

with $D^{1/2} = (exp(-\alpha|\nabla I|^{\beta})\vec{n}\vec{n}^T + \vec{n}^{\perp}\vec{n}^{\perp^T})^2$ and $\vec{n} = \frac{\nabla I}{|\nabla I|}$.

However, this is known to result in an over-segmentation of the displacement field produced by image intensity changes that are not related to real motion discontinuities.

## 4.4 Data term linearization

The data term in the formulation of the variational matching problem contains non-linearities in terms of $u$ and $v$. These non-linearities present important obstacles to minimize the formulated energies. Therefore, we approximate them through a first order Taylor expansion. These linearizations can be done before formulating the problem in the Euler-Lagrange form or after.

### 4.4.1 Early data term linearization

An *early* linearization of the data term of the function 4.1 is presented in equation 4.2. How was it obtained? If we drop the previous notation and replace $I_1(x, y) = I(x, y, t)$ and $I_2(x, y) = I(x + u, y + v, t + 1)$, following the brightness constancy assumption we search for $u$ and $v$ such that $I(x, y, t) = I(x + u, y + v, t + 1)$. This term can be linearized through a first-order Taylor expansion that yields:

$$I(x, y, t + 1) = I(x, y, t) + u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} \tag{4.10}$$

which can be simplified to:

$$\frac{\partial I}{\partial t} + u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} = 0 \tag{4.11}$$

This initial linearization provides the means for an straight forward energy optimization. In the next sub-section we discuss what happens if this is done in a different way.

### 4.4.2 Late data term linearization

It can be argued that the data term linearization needs not to be done in an initial stage and some benefit can be drawn in doing so as a last step. Following [18], equation 4.12 can be directly formulated according to the corresponding Euler-Lagrange equations to obtain the equations 4.13 (with variable replacements formulated in 4.14). The problem with equations 4.13 being that they are still non linear in the robust functions $\Psi(s^2)$ used in the data and regularization terms. Furthermore, they are non linear for $u$ and $v$ in the data term. To overcome the first non-linearity, the terms $\Psi'$ are assumed constant at each step and treated as a delayed non-linearity [18]. To overcome the second non-linearity in the data term, a first order Taylor expansion is

performed and 4.15 is obtained. This approach has been proven to give better results for large displacements when compared to those of early linearization methods [18, 17].

$$E(u,v) = \int\int \Psi(|I_1(x,y) - I_2(x+u,y+v)|^2) + \Psi(|\nabla I_1(x,y) - \nabla I_2(x+u,y+v)|^2) \\ +\alpha\Psi(|\nabla u(x,y)|^2 + |\nabla v(x,y)|^2)dxdy \tag{4.12}$$

Note that in the equation above the same penalizer functions $\Psi$ are used for all three terms but in practice different ones can be used.

$$\begin{aligned} \Psi'(I_t^2)I_tI_x + \beta\Psi'(I_{xt}^2 + I_{yt}^2)(I_{xx}I_{xt} + I_{xy}I_{yt}) - \alpha div(\Psi'(|\nabla u|^2 + |\nabla v|^2))\nabla u) = 0 \\ \Psi'(I_t^2)I_tI_y + \beta\Psi'(I_{xt}^2 + I_{yt}^2)(I_{xy}I_{xt} + I_{yy}I_{yt}) - \alpha div(\Psi'(|\nabla u|^2 + |\nabla v|^2))\nabla v) = 0 \end{aligned} \tag{4.13}$$

with the abbreviations:

$$\begin{aligned} I_x &:= \partial_x I_2(x+u,y+v) \\ I_y &:= \partial_y I_2(x+u,y+v) \\ I_t &:= I_2(x+u,y+v) - I_1(x+u,y+v) \\ I_{xx} &:= \partial_{xx} I_2(x+u,y+v) \\ I_{xy} &:= \partial_{xy} I_2(x+u,y+v) \\ I_{xt} &:= \partial_x I_2(x+u,y+v) - \partial_x I_1(x+u,y+v) \\ I_{yx} &:= \partial_{yx} I_2(x+u,y+v) \\ I_{yy} &:= \partial_{yy} I_2(x+u,y+v) \\ I_{yt} &:= \partial_y I_2(x+u,y+v) - \partial_y I_1(x+u,y+v) \end{aligned} \tag{4.14}$$

$$
\begin{aligned}
\Psi'_D I_x(I_t + I_x u + I_y v) + \beta \Psi'_G I_{xx}(I_{xt} + I_{xx} u + I_{xy} v) + \beta \Psi'_G I_{yx}(I_{yt} + I_{yx} u + I_{yy} v) \\
-\alpha div(\Psi'_S \nabla u) = 0 \\
\Psi'_D I_y(I_t + I_x u + I_y v) + \beta \Psi'_G I_{xy}(I_{xt} + I_{xx} u + I_{xy} v) + \beta \Psi'_G I_{yy}(I_{yt} + I_{yx} u + I_{yy} v) \\
-\alpha div(\Psi'_S \nabla v) = 0
\end{aligned} \tag{4.15}
$$

with the abbreviations:

$$
\begin{aligned}
\Psi'_D &:= \Psi'(|I_t + I_x u + I_y v|^2) \\
\Psi'_G &:= \Psi'(|I_{xt} + I_{xx} u + I_{xy} v|^2 + |I_{yt} + I_{yx} u + I_{yy} v|^2) \\
\Psi'_S &:= \Psi'(|\nabla u|^2 + |\nabla v|^2))
\end{aligned} \tag{4.16}
$$

The data and regularization terms can be formulated in several manners and calibrated for specific tasks. In the next sub-section and without modifying the latter terms, we present two ways to improve the variational matching formulation in the context of 3D reconstruction.

## 4.5 Additional cues from scene geometry

In order to improve classical variational matching methods using information about the scene's geometry, additional conditions can be introduced in the formulation of the problem. In the remainder of this chapter we focus on the use epipolar (e.g. [132]) and feature based constraints (e.g. [19]), which will be detailed in the next sub-sections.

### 4.5.1 Epipolar geometry

Analyzing the matching problem problem in $3D$ space we can obtain additional constrains that will help us solve our system. If we observe figure 4-2 we can perceive

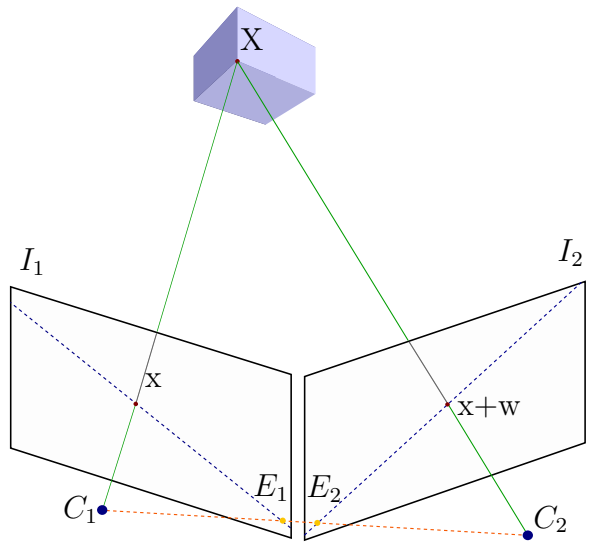



Figure 4-2: Epipolar geometry

that the point X in space is projected into $\mathbf{I}_1$ and $\mathbf{I}_2$ at position $\mathbf{x}_1 = x$ and $\mathbf{x}_2 = x + w$ respectively. These positions are conditioned by the following equation:

$$(x + w(x))^T F x = 0 \tag{4.17}$$

where F represents the fundamental matrix which could be obtained from a set of correspondences [46]. In practice, $\mathbf{F}$ does not relate the points $\mathbf{x}_1$ and $\mathbf{x}_2$ exactly but it should give a close approximation. This constraint is then accounted for as the distance of the point $\mathbf{x}_2$ to the line directed by $F\mathbf{x}_1$ and hence the relation in equation 4.17 can be formulated as a line equation [132]:

$$au + bv + c = 0 \tag{4.18}$$

with $[a, b, c]^T = F[x, y, 1]^T$ and $[u, v] = w$. The objective is then to minimize the distance of the matching points to the epipolar lines.

$$E_e u, v = \int\int \frac{|a(x,y)u(x,y) + a(x,y)v(x,y) + c(x,y)|^2}{a^2 + b^2} dxdy \tag{4.19}$$

### 4.5.2 Sparse feature matches

Few papers, including [109, 19], have introduced the use of descriptors in the variational formulation of the dense matching problem. For example, [19] proposes to obtain dense feature descriptors, match them and include these pre-computed matches as driving points. Here we illustrate the introduction of sparse feature detection and matching as a constraint using the SIFT [74] or LDAHASH [20] features but any other could be used. The features are computed and filtered using RANSAC [46, 31] to use only the highly reliable matches that comply with scene geometry. If the parameters of the cameras are known, one can directly use that information filter incorrect feature matches in a more reliable and fast way. This filtering allows introducing the feature matches with a strong weight in the variational formulation, contrary to [19] where points are only matched using optical information and, are obtained densely, introducing possible mismatches.

We denote the introduced features term as:

$$E_f(u,v) = \sum_i |u_i - u_i^f|^2 + |v_i - v_i^f|^2 \tag{4.20}$$

where $i$ spans the range of matching feature points and $w_i^f$ represents the prescribed displacement.

## 4.6 Solving the variational matching problem

If we take equation 4.15 and add the terms 4.19 and 4.20 we obtain a variational matching formulation that takes advantage of scene geometry. These two terms have to be added following the euler-lagrange form:

$$S(E[u,v]) = \int E(x,y,u,v,u_x,u_y,v_x,v_y)dx \tag{4.21}$$

to find the stationary points $[u,v]$ using the following two equations:

$$\frac{\partial E}{\partial u} - \left(\frac{\partial}{\partial x}\frac{\partial E}{\partial u_x} + \frac{\partial}{\partial y}\frac{\partial E}{\partial u_y}\right) = 0 \tag{4.22}$$

$$\frac{\partial E}{\partial v} - \left(\frac{\partial}{\partial x}\frac{\partial E}{\partial v_x} + \frac{\partial}{\partial y}\frac{\partial E}{\partial v_y}\right) = 0 \tag{4.23}$$

The epipolar and feature terms formulated according to the equations above are given by:

$$\begin{aligned} \gamma\Psi'_e\frac{a(au+bv+c)}{a^2+b^2} + \phi(u(x,y)-u_f(x,y)) = 0 \\ \gamma\Psi'_e\frac{b(au+bv+c)}{a^2+b^2} + \phi(v(x,y)-v_f(x,y)) = 0 \end{aligned} \tag{4.24}$$

with $\Psi'_e = \Psi'((au+bv+c)/(a^2+b^2))$ and $\phi$ taking the value of zero for the locations where features are not available.

Note that all $\Psi'$ terms in 4.24 and 4.15 come from the robust penalizers used. They still result in a non-linear term for which [18] proposes to treat them as a delayed

non-linearity that is estimated and fixed iteratively. A slightly different solution is given in [12] where gradual non-convexity is used to solve for this term. Once the linearization is performed, we can move on to minimize the energy.

A variety of options are available to solve for the equations formulated above. For example, one can use the Jacobi, Gauss-Seidel, Conjugate Gradient or, more preferably, Successive Over Relaxation (SOR) methods. They are based on an iterative process that updates $[u, v]$ by small increments $[du, dv]$. Therefore, the problem can be formulated as an optimization of $E(u + du, v + dv)$, where $du$ and $dv$ are treated as unknowns.

### 4.6.1 Successive over relaxation: SOR

If we formulate the optical flow problem as a system of the form $Ax = b$, we can use the Gauss-Seidel method. In this method, one can write an iterative procedure that updates $x$ as $x^{k+1} = (D - L)^{-1}(b + Ux^k)$, with $D$ the diagonal part of the of the matrix $A$, $L$ the lower triangular part of $A$ and $U$ the upper triangular part of $A$. By doing this, we can take advantage of the triangular form of $L$ and $x^k + 1$ can be obtained sequentially following:

$$x_i^{k+1} = \frac{1}{a_{ii}} \left( b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^k \right) \tag{4.25}$$

Successive over relaxation presents a variant of Gauss-Seidel that achieves faster convergence rates by performing point-wise extrapolation. The SOR formulation works by also decomposing $A = D + L + U$ as for Gauss-Seidel, but it formulates the iterative process as: $x^{k+1} = (D - \omega L)^{-1}(\omega b - [\omega U + (\omega - 1)D]x^k)$. Next, using the triangular form of $D + \omega L$ we can write the iterations as in equation 4.26.

$$x_i^{k+1} = (1-\omega)x_i^k \frac{\omega}{a_{ii}} \left( b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^k \right) \tag{4.26}$$

The parameter $\omega$ represents a relaxation factor that is usually obtained through by performing a parameter exploration. It is dependent on the properties of the matrix $A$. Also, if $A$ is a symmetric and positive-definite matrix, convergence is guaranteed for $0 < \omega < 2$ [95]. Using SOR for the computation of the optical flow can result in an acceleration in the convergence rate of one to two orders of magnitude. Common values for $\omega$ used in classical optical flow formulations vary between 1.6 and 1.9. Note that SOR is equivalent to Gauss-Seidel when $\omega = 1$.

The SOR technique has been widely used to solve variational matching problems (e.g. [18, 121]) because of it's simplicity and performance. Nevertheless, in the next sub-section we outline Alternating Direction Implicit; an alternative technique that presents similar performance.

### 4.6.2 Alternating direction implicit: ADI

ADI methods are commonly used to solve for heat conduction problems and diffusion equations. They represent a simple example of operator splitting methods and where introduced by Peaceman and Rachford [98] to solve time-dependant heat equation in two dimensions. It works by solving the problem in the 2D grid line by line, first in one direction for all lines and then in the other direction; hence the name alternating direction. Starting from the same problem form as before $Ax = b$, the first step in ADI is to decompose $A = H+V+S$. $H$ represents the $x$-derivative term, $V$ represents the $y$-derivative term and $S$ represents the zero order term so:

$$(H + S + V)x = b \tag{4.27}$$

From the previous equation we can formulate an iterative scheme expressed as:

$$(I + H + \theta S)x = (I - V - (1 - \theta)S)x + b \tag{4.28}$$

Setting $\theta = 1/2$ and introducing an iterator factor $r$ that multiplies $I$ in 4.28 we obtain the horizontal and vertical iterators:

$$(rI + H + \frac{S}{2})x' = (rI - V - \frac{S}{2})x^k + b \tag{4.29}$$

$$(rI + V + \frac{S}{2})x^{k+1} = (rI - H - \frac{S}{2})x' + b \tag{4.30}$$

with $x'$ introduced as an intermediate result. The value of $r$ can be difficult to estimate but it should be greater than zero to converge.

What is important in this method is that each line computed using equations 4.29 and 4.30, is independent to each other. Therefore the Thomas tri-diagonal method [24] can be used to solve each step.

### 4.6.3 Coarse to fine

The first-order approximation of the data term in the energy $E(u+du, v+dv)$ allows only for matches to be found if the solution falls in one of the surrounding pixels. One important approach to overcome this limitation is to solve the problem in a coarse to fine manner. In this way, large displacements in the full scale problem, will be seen as very small displacements in the coarser scales. This approach starts by building image pyramids with increasing blurring and downsampling [76, 12, 18]. In this approach, displacement vectors are initialized to zero and we first compute $[du^0, dv^0]$ for the top level of the pyramid (coarsest representation). Then $[u^0, v^0]$ are updated, $[u^0, v^0] = [u^0, v^0] + [du^0, dv^0]$ and up-sampled to obtain $[u^1, v^1]$. This

up-sampled version is used to warp the moving image in the next level of the pyramid ($I_2^1$) so that the current estimation is closer to the result [11]. This process continues until the finest level of the pyramid is resolved.

In essence, this procedure makes large strides towards the final result at the top levels of the pyramid. This represent an important speedup since top levels of the pyramid have few unknowns and, therefore, iterations are cheaper to perform. Moreover, and directly related to the first order linearization of the data term, the top levels of the pyramid hold images with lower frequency components, which make the optimization procedure less susceptible to stuck in local minimums. More elaborated multi-scale approaches are presented in the next subsection.

### 4.6.4 Multi-grid

The coarse-to-fine technique outlined above slow down its convergence after a few iterations. Muti-grid methods [15] aim to speedup the convergence rate by obtaining large updates from coarse grids. The image pyramid previously presented (in 4.6.3) formulates a simple way of multi-grid with a coarse to fine approach, also called cascadic multi-grid. More sophisticated multi-grid methods propose to go up and down in scales, following $V$ and $W$ shaped patterns. This is approach is taken since few iterations on higher resolutions can bring important information earlier in the problem.

For example, representing by $f$ a fine scale and by $c$ a coarser level, a $V$ shaped linear multi-grid solver can follow the next steps:

1. Solve for high frequency errors (smooth)
   (a) Apply Gauss-Seidel solver to $A^f x^f = b^f$
   (b) Compute residual $r^f = b^f - A^f \hat{x}^f$
   (c) Obtain the error equation $A^f e^f = r^f$

(d) Perform pixel averaging over $e^f$ to obtain $x^c$

(e) Rediscretize euler-lagrange equations of $A^f$ to obtain $A^c$

(f) Now we can transfer $A^f e^f = r^f \rightarrow A^c x^c = r^c$

2. Solve for low frequency errors (restrict)

   (a) Solve $A^c x^c = r^c$ using SOR

   (b) Perform pixel interpolation over $x^c$ to obtain new $e^f$

3. Correct fine grid with coarse grid results

   (a) Correct approximation $x^f = \hat{x}^f + e^f$

4. Smooth again

   (a) Apply Gauss-Seidel solver to $A^f x^f = b^f$

## 4.7 Experiments

We evaluate the performance of several established variational matching (optical flow) algorithms in the context of wide-baseline setups. The selected scenes present common cases that include untextured regions, sharp features, occlusions, small details and large distortions. The evaluation is performed to the algorithms presented in: Brox & Malik (*Brox11*) [19], Sun et al.(*Sun*) [121], Wedel et al.(*Wedel*) [136], Brox et al.(*Brox04*) [18]. Additionally, we compare the variational results with a global discrete algorithm Liu et al.(*Liu*) [72], a quasidense propagation matching algorithm: Kannala & Brandt (*Kannala*) [54], and one state of the art algorithm for 3D model acquisition: Furukawa & Ponce (*Furukawa*) [35]. These last two methods will be explained in detail in Chapter 6. In all cases implementation provided by the authors was used.

### 4.7.1 Datasets

In this evaluation, we use pictures of four scenes in non controlled environments. Each scene is computed in five different setups that present an increase of angle with respect to a reference frame. First, we evaluate on the frames two to seven of the *Fountain* dataset [120]. This scene includes ground truth and it presents a widebaseline problem with large distortions, sharp features and occluded regions. Next, the *Dino* dataset [111], frames 26 to 31 (no ground-truth). This scene presents untextured regions. The third evaluation is performed on our own *Face* dataset, which comprises a set of face images with no ground-truth. Speckle was added to the face to provide more descriptive texture. The last evaluation is performed on our own dataset of a *Box* with repetitive patterns (no ground-truth). See figures 4-3 to 4-6.

Figure 4-3: Fountain scene. Views 2 and 7

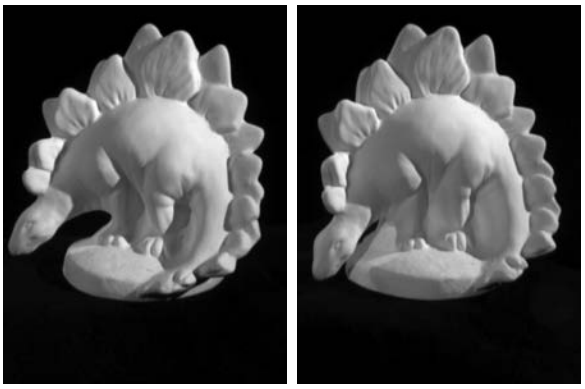


Figure 4-4: Dino scene. Views 26 and 31

Figure 4-5: Face scene. Views 1 and 6

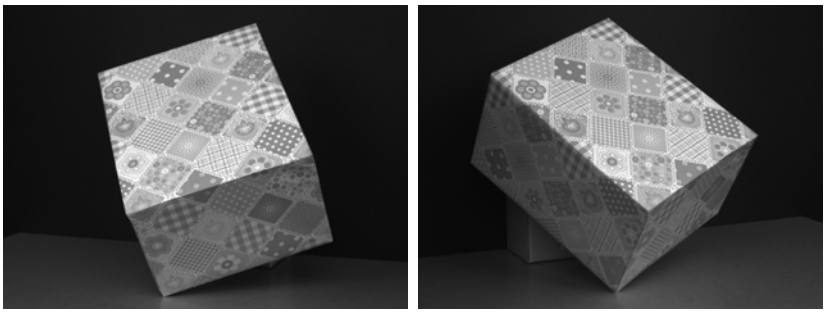


Figure 4-6: Box 1 with repetitive pattern . Views 1 and 6

### 4.7.2 Methods and criteria

We evaluate the results on the *Fountain* scene based on the average end-point error (AEP) commonly used in optical flow. For this case the 2D matching ground-truth is computed by projecting the 3D structure onto each image.

For all the scenes, the method's results are evaluated based on their average distance to the epipolar line (ADE) and the distance of several precomputed and validated sparse matches (ADF). The matches are computed using the LDAHASH 3.3.3 detector and filtered with RANSAC. In addition, we include the standard deviations (SD) of each error.

For the computation of the endpoint error and epipolar error in the fountain scene a mask is used to evaluate only for the regions projected inside the used views. For the Face, Box and Dino scenes masks are used to evaluate only for the objects without the background.

We define the average endpoint error (equation 4.31) as in [96, 6]

$$AEP = \frac{1}{n}\sum_{i=1}^{n}\sqrt{([u_i^0 - u_i]^2 + [v_i^0 - v_i]^2)} \tag{4.31}$$

The average distance to the epipolar line (ADE) is defined by 4.32

$$ADE = \frac{1}{n}\sum_{i=1}^{n}\frac{|a_i x_i' + b_i y_i' + c_i|}{\sqrt{a_i^2 + b_i^2}} \tag{4.32}$$

where $[a, b, c] = F \times X$ is the equation of the epipolar line defined for each point $X = [x, y, 1]$ in $I_1$ and n is the number of pixels in $I_1$

We define the average distance to the features (ADF) in equation 4.32

$$ADF = \frac{1}{m}\sum_{i=1}^{m}\sqrt{([u_i^f - u_i]^2 + [v_i^f - v_i]^2)} \tag{4.33}$$

where m represents the number of features, $u_i^f$ and $v_i^f$ represent respectively the $x$ and $y$ displacement estimated for the $i^{th}$ pre-computed feature. In the following results sub-section the setups of the experiments (angles in degrees) and the number of features used for each evaluation are presented along with their respective results.

The following evaluations include a modified version of Brox et al. [18] that aims to achieve better results in the case of the wide-baseline setups. The modifications consist on the inclusion of an epipolar constraint (see subsection 4.5.1) and feature constraints (see subsection 4.5.2). We will call this method *Brox+EF*. The features used are LDAHASH, which are filtered using RANSAC. The features are included in the optimization with a high weight since they are already geometrically filtered, which is not the case for [19]. The epipolar term is introduced as a distance to the epipolar line defined for each pixel (see 4.5.1).

### Evaluation results

Tables 4.1 to 4.9 present the results obtained with different variational matching methods at increasingly wider baseline setups. A result of flow equal to zero is included in the figures (0 flow) to work as a reference of performance and validate the increasing baseline setups. First, we evaluated the methods on the Fountain scene which includes ground truth to compare to. Tables 4.1 and 4.3 and, more clearly, figures 4-9 and 4-10 confirm the correlation between the average endpoint error and the average distance to features, which is important for the validity of our tests. We can also infer a lower correlation between average distance to the epipolar line and the average endpoint error for the global models.

The method Brox+EF achieves the best result for the global methods tested although, it presents a higher average endpoint error than Kannala & Brandt. With the wider-baseline setups the errors increase noticeably. In particular, the methods of Wedel et al. and Sun et al., perform well for the narrow baseline cases and dramatically decrease their performance as the setup turns wider. In a similar way, the

performance of Brox04 reduces for the cases of wider baselines but the method of Liu et al. proves more robust; which displays the importance of descriptor matching for wide baselines. In figure 4-8 we present the percentage of points in the visible surface that have an endpoint error of one pixel or less. In this figure we observe that the method Brox+EF obtains more accurate results than any other of the variational methods tested. It also returns more accurate points for the wide-baseline compared to that of the global and region-based methods. Note that despite being clear in figure 4-8 that our method Brox+EF obtains more accurate points in the wide-baseline case, the average of the endpoint error (and it's standard deviation) is close to that of the region based models. It is clear that the advantage of the region based methods is the ability to reject bad matches. For a better visualization, we present in figure 4-7 the distribution of the endpoint error in the scene, taking frame 2 as a reference.

The methods that include descriptors, namely Liu et al. and Brox & Malik (*Brox11*), present the lower performance in the face scene. In fact, the texture added into a non-planar object presents a more difficult problem for the descriptor matching method. In more detail, in figure 4-14 we can observe that these two methods yield a large distance to the epipolar line when compared to the rest of the methods. On the other hand, for the box scene with flat surfaces, the feature based algorithms perform better than the rest of the global methods.

Overall, in our experiments all methods considerably reduce its performance beyond the 30 degrees separation angle between cameras. Note that for the method of Kannala & Brandt and Furukawa & Ponce, the average distance to the features is not included in the results since the features can be used as initial seeds in the algorithms.

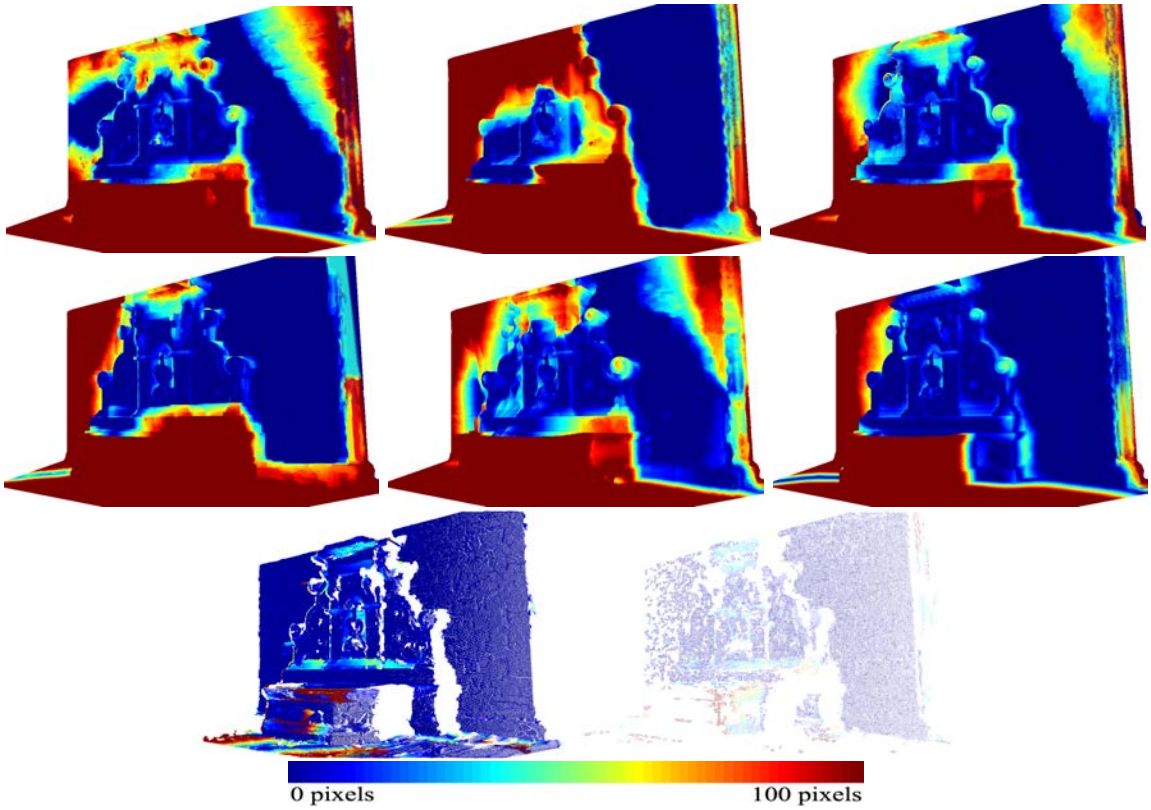



Figure 4-7: Endpoint error results for the fountain scene, view 2-5. From left to right and top to bottom; Brox04, Brox11, Sun, Liu, Wedel, Brox04+EF, Kannala, Furukawa. On the bottom, the error color map with dark red representing distances of one hundred pixels or more

| Cam | Angle | Brox11 | Sun | Wedel | Liu | Brox04 | Brox04+E+F | Kannala | Furuk. |
|---|---|---|---|---|---|---|---|---|---|
| 2-3 | 10.66 | 9.462 | 8.214 | 9.813 | 7.526 | 7.996 | 5.101 | 2.875 | 2.143 |
| 2-4 | 21.12 | 45.877 | 29.511 | 33.287 | 37.006 | 29.851 | 13.833 | 4.827 | 3.215 |
| 2-5 | 32.4 | 74.382 | 55.559 | 60.914 | 69.404 | 61.461 | 40.082 | 11.629 | 4.870 |
| 2-6 | 42.32 | 120.97 | 108.81 | 159.06 | 101.86 | 125.94 | 47.558 | 18.662 | 5.165 |
| 2-7 | 53.52 | 150.49 | 211.42 | 186.17 | 135.17 | 155.81 | 51.778 | 57.269 | 21.346 |

Table 4.1: Average end point error in the fountain scene

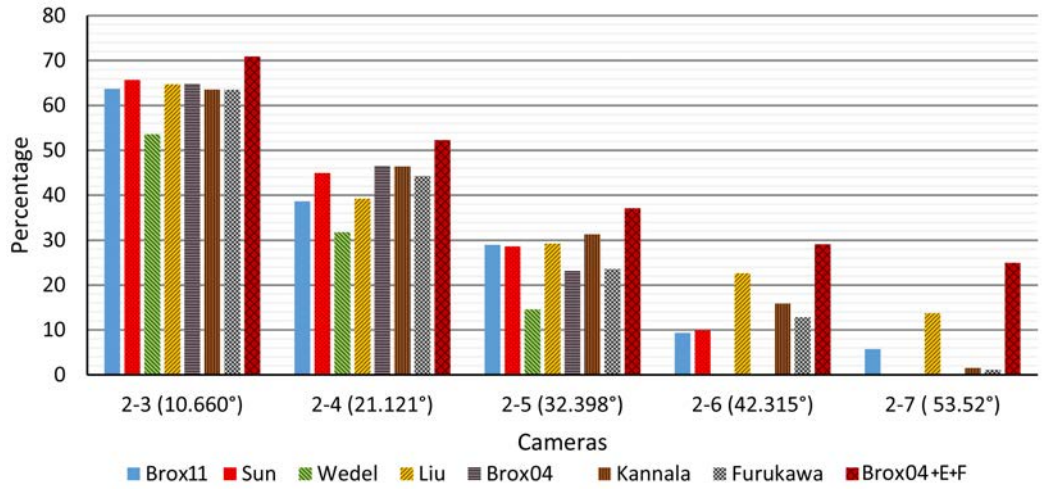



Figure 4-8: Percentage of endpoint error less or equal to 1 pixel in the fountain scene

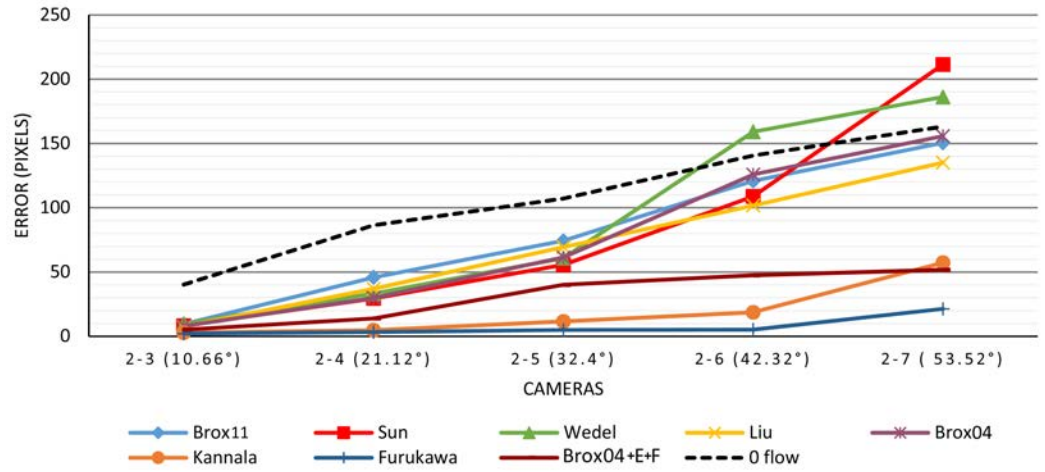



Figure 4-9: Average endpoint error in the fountain scene

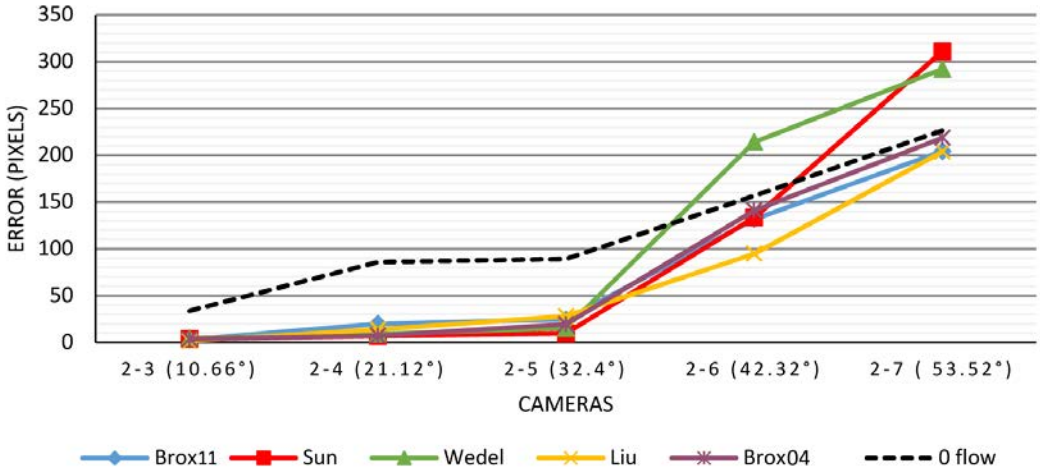



Figure 4-10: Average distance to features in the fountain scene

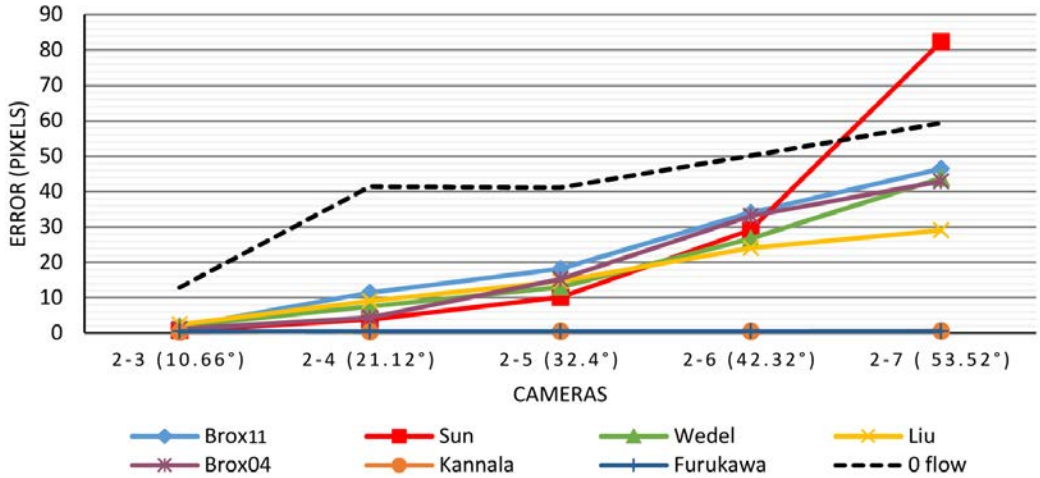



Figure 4-11: Average distance to the epipolar line in the fountain scene

| Cam | Angle | Brox11 | Sun | Wedel | Liu | Brox04 | Kannala | Furuk |
|---|---|---|---|---|---|---|---|---|
| 2-3 | 10.66 | 1.723 | 0.817 | 1.943 | 2.456 | 1.153 | 0.340 | 0.441 |
| 2-4 | 21.12 | 11.406 | 3.820 | 7.513 | 9.069 | 4.368 | 0.419 | 0.470 |
| 2-5 | 32.4 | 18.178 | 10.178 | 12.987 | 14.672 | 15.233 | 0.507 | 0.496 |
| 2-6 | 42.32 | 33.977 | 29.175 | 26.639 | 24.046 | 33.093 | 0.508 | 0.501 |
| 2-7 | 53.52 | 46.432 | 82.276 | 43.797 | 29.057 | 42.896 | 0.583 | 0.508 |

Table 4.2: Average distance to the epipolar line in the fountain scene

| Cam | Angle | Brox11 | Sun | Wedel | Liu | Brox04 |
|---|---|---|---|---|---|---|
| 2-3 | 10.66 | 2.912 | 3.584 | 4.334 | 1.409 | 3.565 |
| 2-4 | 21.12 | 19.840 | 6.756 | 9.049 | 13.775 | 6.980 |
| 2-5 | 32.4 | 24.477 | 9.686 | 15.135 | 28.010 | 19.453 |
| 2-6 | 42.32 | 131.69 | 133.48 | 214.31 | 94.75 | 141.07 |
| 2-7 | 53.52 | 204.29 | 311.02 | 291.96 | 203.72 | 218.61 |

Table 4.3: Average distance to features in the fountain scene

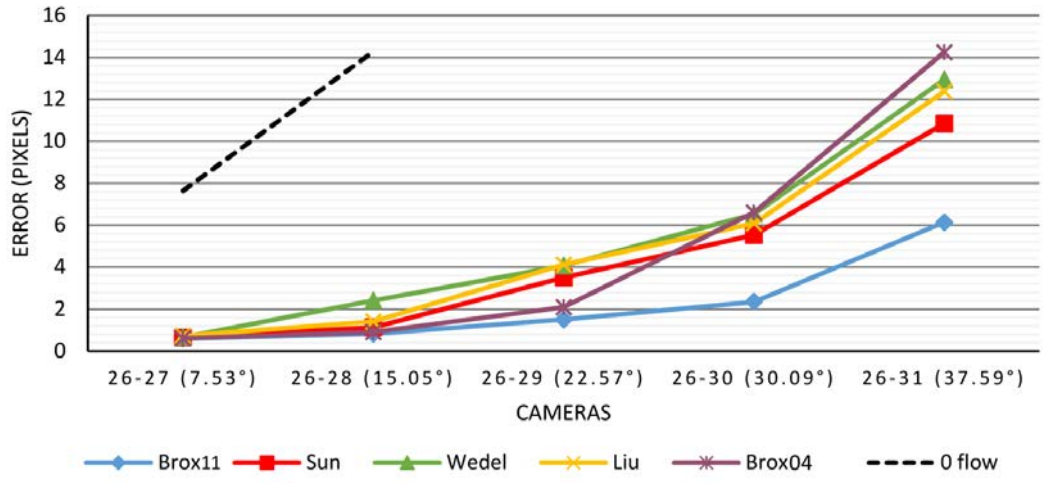



Figure 4-12: Average distance to features in the dino scene

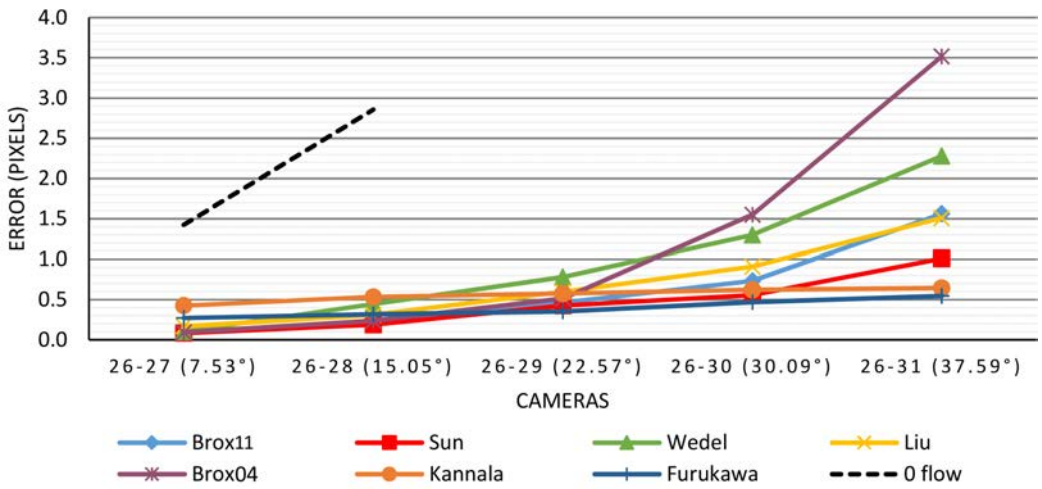



Figure 4-13: Average distance to the epipolar line in the dino scene

| Cam | Angle | Brox11 | Sun | Wedel | Liu | Brox04 | Kannala | Furuk. |
|---|---|---|---|---|---|---|---|---|
| 26-27 | 7.53 | 0.092 | 0.083 | 0.104 | 0.167 | 0.093 | 0.426 | 0.271 |
| 26-28 | 15.05 | 0.193 | 0.186 | 0.444 | 0.309 | 0.239 | 0.533 | 0.318 |
| 26-29 | 22.57 | 0.464 | 0.425 | 0.781 | 0.601 | 0.514 | 0.578 | 0.352 |
| 26-30 | 30.09 | 0.734 | 0.555 | 1.298 | 0.909 | 1.550 | 0.624 | 0.469 |
| 26-31 | 37.58 | 1.559 | 1.009 | 2.279 | 1.508 | 3.518 | 0.646 | 0.546 |

Table 4.4: Average distance to the epipolar line for the dino scene

| Cam | Angle | Brox11 | Sun | Wedel | Liu | Brox04 |
|---|---|---|---|---|---|---|
| 26-27 | 7.53 | 0.590 | 0.641 | 0.660 | 0.725 | 0.613 |
| 26-28 | 15.05 | 0.827 | 1.136 | 2.410 | 1.416 | 0.909 |
| 26-29 | 22.57 | 1.511 | 3.503 | 4.076 | 4.120 | 2.085 |
| 26-30 | 30.09 | 2.350 | 5.544 | 6.539 | 6.089 | 6.604 |
| 26-31 | 37.58 | 6.139 | 10.854 | 12.948 | 12.378 | 14.252 |

Table 4.5: Average distance to features for the dino scene

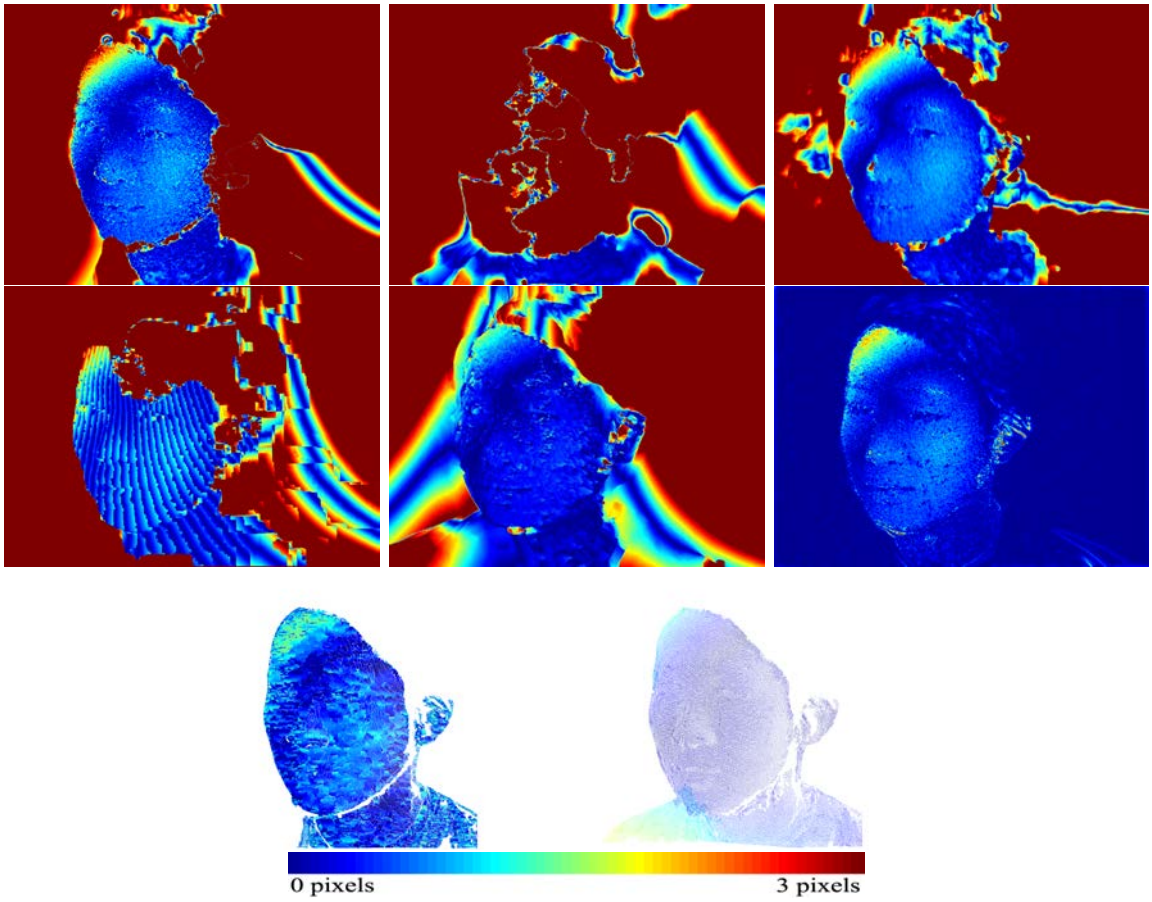



Figure 4-14: Distance to the epipolar line results for the face scene, views 1-2. From left to right and top to bottom: Brox04, Brox11, Sun, Liu, Wedel, Brox+E+F, Kannala, Furukawa. On the bottom, the error color map with dark red representing distances of three pixels or more

| Cam | Angle | Brox11 | Sun | Wedel | Liu | Brox04 | Kannala | Furuk |
|---|---|---|---|---|---|---|---|---|
| 1-2 | 9.22 | 6.453 | 1.405 | 0.366 | 2.453 | 0.593 | 0.555 | 0.761 |
| 1-3 | 20.40 | 7.713 | 5.098 | 4.007 | 10.041 | 5.551 | 0.600 | 0.750 |
| 1-4 | 28.30 | 11.970 | 11.179 | 11.183 | 12.197 | 8.718 | 0.602 | 0.753 |
| 1-5 | 39.78 | 17.489 | 22.158 | 10.424 | 16.140 | 20.326 | 0.654 | 0.942 |
| 1-6 | 48.53 | 18.402 | 25.403 | 24.623 | 19.219 | 22.368 | 0.730 | |

Table 4.6: Average distance to the epipolar line for the face scene

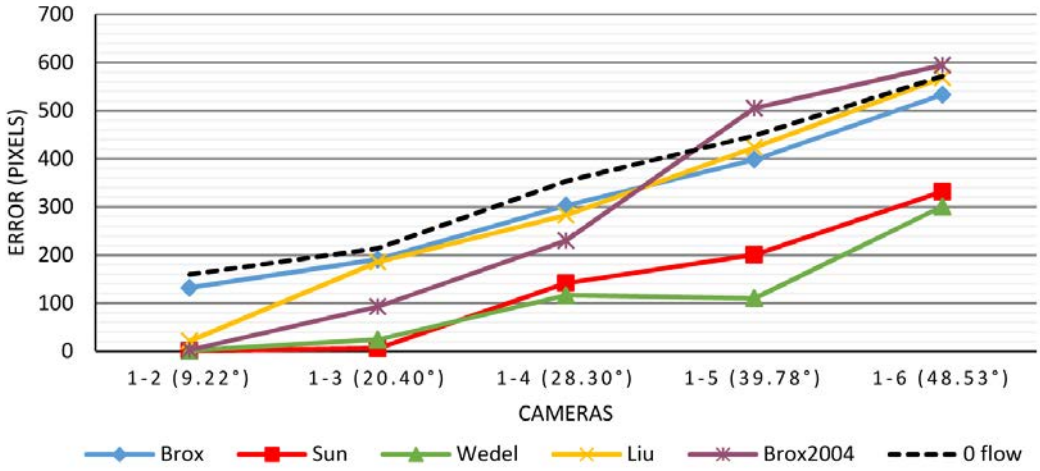



Figure 4-15: ADF in the face scene

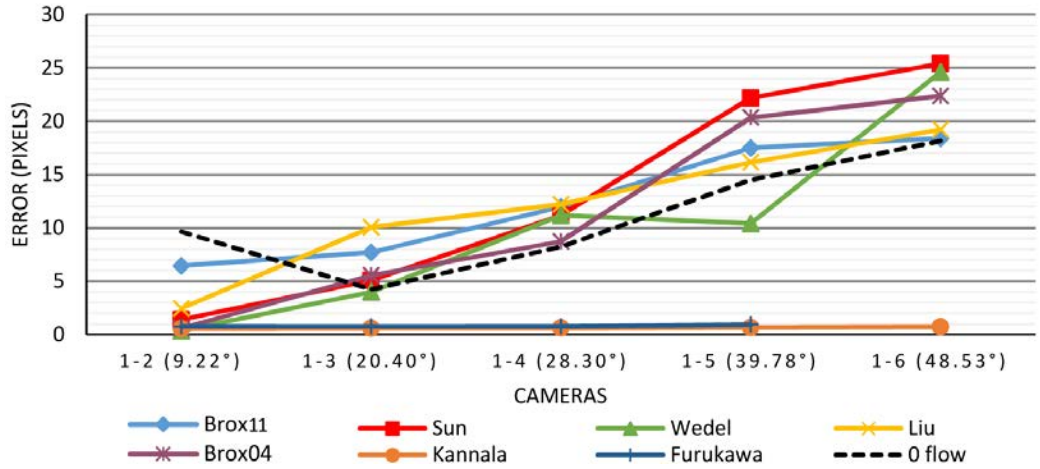



Figure 4-16: ADE in the face scene

| Cam | Angle | Brox11 | Sun | Wedel | Liu | Brox04 |
|---|---|---|---|---|---|---|
| 1-2 | 9.22 | 132.190 | 1.350 | 1.346 | 21.149 | 2.596 |
| 1-3 | 20.40 | 190.51 | 5.852 | 23.906 | 185.27 | 92.73 |
| 1-4 | 28.30 | 302.61 | 141.89 | 116.35 | 282.83 | 230.00 |
| 1-5 | 39.78 | 398.45 | 200.29 | 110.07 | 423.33 | 505.06 |
| 1-6 | 48.53 | 532.92 | 331.95 | 300.99 | 568.34 | 594.51 |

Table 4.7: Average distance to features for the face scene

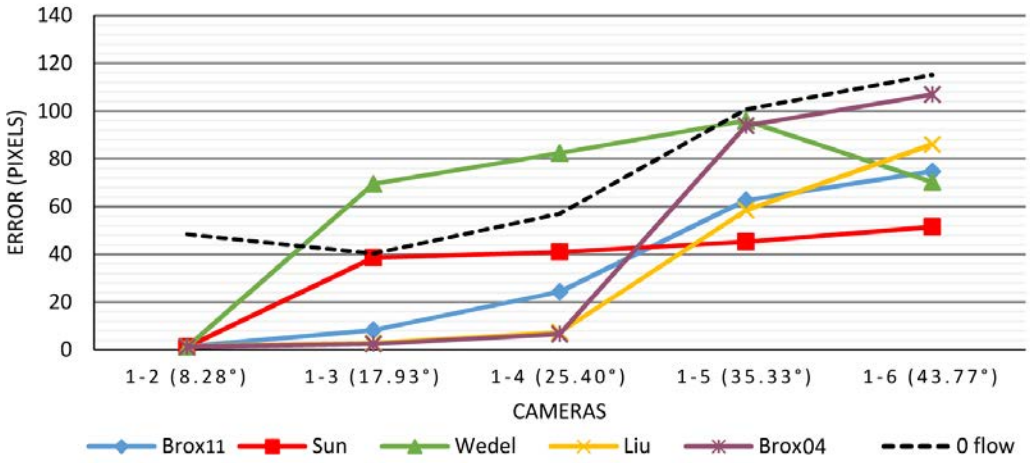



Figure 4-17: Average distance to features in the box scene

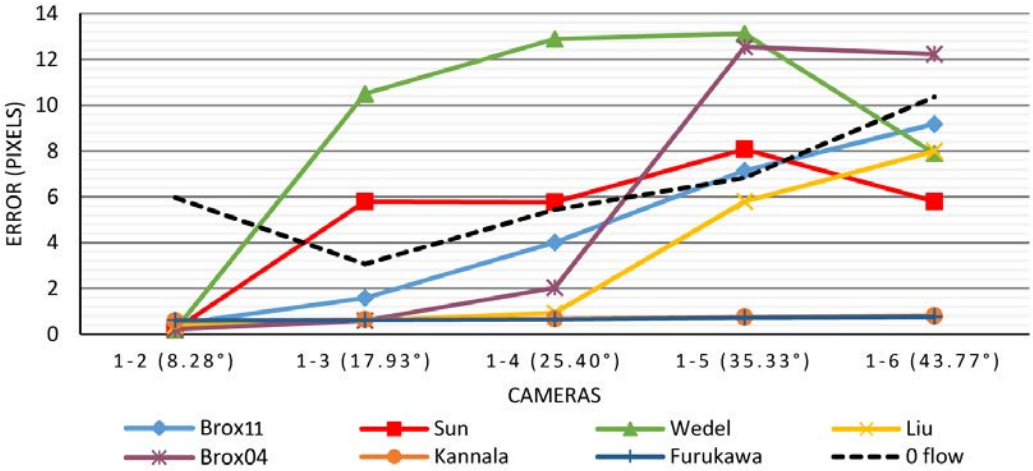



Figure 4-18: Average distance to the epipolar line in the box scene

| Cam | Angle | Brox11 | Sun | Wedel | Liu | Brox04 | Kannala | Furuk. |
|---|---|---|---|---|---|---|---|---|
| 1-2 | 8.28 | 0.433 | 0.201 | 0.202 | 0.388 | 0.218 | 0.590 | 0.610 |
| 1-3 | 17.93 | 1.588 | 5.802 | 10.501 | 0.606 | 0.596 | 0.646 | 0.613 |
| 1-4 | 25.40 | 4.012 | 5.771 | 12.892 | 0.943 | 2.035 | 0.673 | 0.648 |
| 1-5 | 35.33 | 7.143 | 8.084 | 13.130 | 5.799 | 14.548 | 0.760 | 0.721 |
| 1-6 | 43.77 | 9.185 | 5.806 | 7.901 | 7.999 | 12.231 | 0.802 | 0.763 |

Table 4.8: Average distance to the epipolar line for the box scene

| Cam | Angle | Brox11 | Sun | Wedel | Liu | Brox04 |
|---|---|---|---|---|---|---|
| 1-2 | 8.28 | 1.555 | 1.327 | 1.321 | 1.575 | 1.361 |
| 1-3 | 17.93 | 8.241 | 38.602 | 69.467 | 2.985 | 2.481 |
| 1-4 | 25.40 | 24.346 | 40.962 | 82.50 | 7.403 | 6.583 |
| 1-5 | 35.33 | 62.707 | 45.290 | 95.978 | 58.574 | 93.892 |
| 1-6 | 43.77 | 74.753 | 51.495 | 70.325 | 86.093 | 106.96 |

Table 4.9: Average distance to features for the box scene

# Chapter 5

# Distortion driven variational multi-view reconstruction

The methods covered in the previous chapter achieve, in general, relatively accurate and complete results. However two important issues are identified, pertaining to matching and view merging: i) regions with low visibility and relatively high depth variation are only resolved by the sole regularizer contribution. This often induces wrong matches which tend to bleed into neighboring regions, and more importantly distort nearby features. ii) small matching errors can lead to overlapping surface layers which cannot be easily addressed by standard outlier removal techniques. In both scenarios, we rely on the analysis of the distortion of $3D$ and $2D$ maps in order to improve the quality of the reconstruction. At the matching level, an anisotropic diffusion driven by the $3D$ grid distortion is proposed to steer grid lines away from those problematic regions. At the merging level, advantage is taken of Lambert's cosine law to favor contributions from image areas where the cosine angle between the surface normal and the line of sight is maximal. Tests on standard benchmarks suggest a good blend between computational efficiency, ease of implementation, and reconstruction quality.

## 5.1 Introduction

The classical variational matching, e.g. [3, 146, 119, 118] provides a simple and straightforward mean for multi-view reconstruction. However, the global nature of the solution raises several challenges at the data exploitation and post processing. It is therefore not surprising that on benchmarks such as [120], none of the top performing methods, e.g. [34, 48, 39, 131, 125, 49] is purely variational.

The contributions of this paper to the classical variational setting are twofold. First, a modification of the variational objective function to account for weakly resolved regions is proposed. Second, a filtering approach which allows selecting the best relative contribution of image pairs and triplets is proposed, thus reducing data redundancy and noise effects without sacrificing completeness. The proposed methods are motivated by two observations:

**i) At the matching level**, we observe that the quality of the results deteriorates near large discontinuities where images do not provide enough information. Typical examples are shown in figure 5-1. In fact, the classical variational formulation comprises a data term and a robust smoothing regularizer. So when there is little image information only the smoothing term accounts for the results, as for instance, the basin of the fountain (figure 5-1). While such data points can be filtered out in a post-processing step, the global nature of the solution causes these wrong matches to bleed into neighboring areas, e.g. the rim of the fountain in figure 5-1.

It can be argued that one could possibly tweak the parameters for individual cases to improve the results around the mentioned regions or, use a total variation formulation with an $L1$ norm (e.g. [144]), in which case the results in other regions would be negatively affected by the staircaising effect commonly present in results that use this method [107]. Alternatively, occlusion detection (e.g. [2, 140]) and confidence measures(e.g. [60, 77]) can be included in the variational matching. These approaches have shown improvements for scenes with small discontinuities (e.g. [2, 140, 60]) and complete occlusions, as in [77]. In the later work, occluded regions are excluded from

further computations as soon as they are detected. As a general remark, most of these methods rely only on $2D$ information without accounting for the resulting $3D$ configuration. Furthermore, we are not aware of any special treatments for fixing mismatches located next to largely sheared or occluded regions. The method proposed herein targets the more general problem of resolving regions with highly crammed correspondences regardless if they stem from occlusions or slanted surfaces.

**ii) At the merging level**, we note that due to the dense nature of the solution, the size of the data can become very large, encompassing tens or hundreds of millions of points. The processing of such large $3D$ data is generally a daunting task and raises many challenges. Furthermore, the variational nature of the approach leads to over- and inter-layered data and none of the available methods are specifically tailored to take advantage of its nature.

A typical example of over and interlaying data is shown in figure 5-2.

Certainly, ideas outlined in local matching approaches e.g. [34, 125] and the more general [48, 135] can help filter out the resulting point cloud by means of $3D$ visibility constraints. In particular [135], formulates the problem as quasi-dense matching followed by a global optimization where the number of intersections of the line of sight with the surface is minimized. Other multi-view approaches, which work on the fusion of depth maps to provide an effective multi-view solution (e.g. [145]), have proven successful when used in a small number of images and/or at low resolutions (e.g. [111]); but do not scale well due to the multiple volumetric data structures needed at all times. As a general remark methods which operate globally e.g. [48, 145] have a significant memory footprint which can be challenging when dealing with large datasets or high-resolution images.

In order to address the issues raised at both levels, we rely on the analysis of the distortion of $3D$ and planar maps. We characterize the distortion induced by these maps in order to guide the variational matching and the merging of the results.

At the matching stage, we observe that regions with large discontinuities present important shearing in the resulting 3D grid. In section (5.2), we show how this shearing information can be used to identify problematic regions and how it can be injected into the variational formulation to drive the optimization process adaptively. At the merging stage, described in section (5.3), we regard the matching as a mapping of a regular grid from one view to another and we study the evolution of certain local geometric measures. While we are well aware that thresholds on triangle quality have been commonly used in removing possible outliers, e.g. [48], we emphasize that a naive thresholding indiscriminately removes data and can negatively affect completeness. Moreover, it does not necessarily reduce the over- and inter-laying in the data shown, e.g. figure 5-2.

Most closely related to our work is the method presented in [36] which maintains a constant error while merging multiple views; however, our goal is to define what is best viewed in an image triplet or in an image pair relative to their neighboring views. By recalling that the amount of a viewed area decreases as it is tilted with respect to the view point (see section 5.3), our approach favors matches from regions where the cosine of the surface normal and line of sight is large. Certainly, our merging method does not aim to provide a globally optimal solution, rather, it aims to locally and adaptively select the best contributions from pairs and triplets of views in order to avoid the layering problem without sacrificing completeness or modifying the data.

## 5.2 Geometry driven variational matching

### 5.2.1 Classical variational matching

The variational stereo pair matching formalism emerged from the related optical flow problem which has received extensive attention in the literature [3, 146, 18]. Within this formalism several issues can be addressed including large displacements,

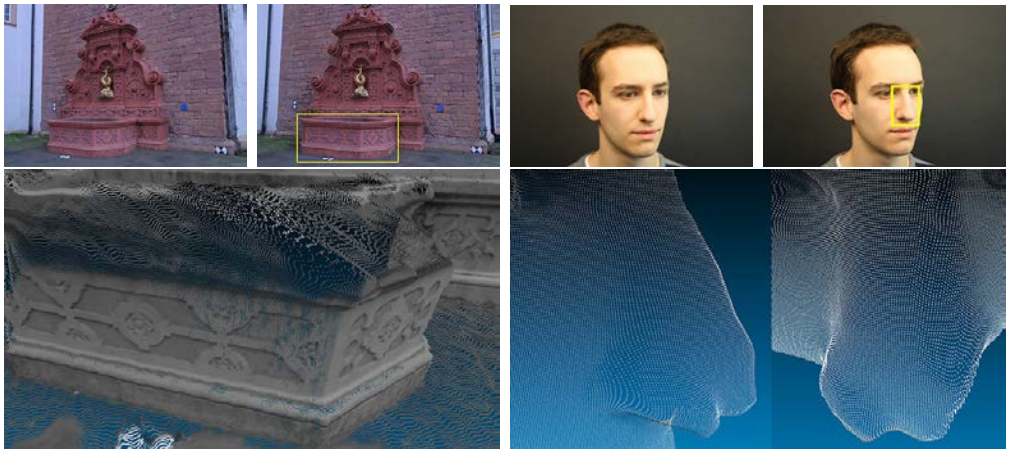


Figure 5-1: A zoom on the reconstruction results of classical variational matching (bottom) across a pair of views from the Fountain-P11 [120] and our face1 data sets (top). Please note, the deterioration along the rim of the fountain and the tip of the nose.

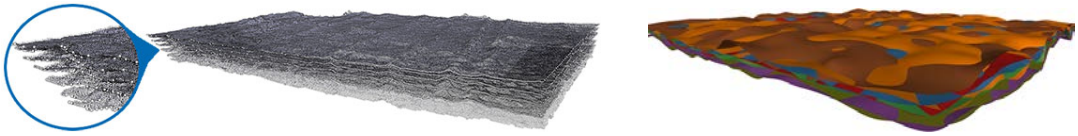


Figure 5-2: Two slices through the fountain back-wall using the classical formulation of variational multi-view reconstruction. Besides from the overwhelming data size, slight matching errors can lead to over- (left) and inter- (right) laying data.

illumination variations, and strong and large discontinuities. It aims to minimize the following objective function:

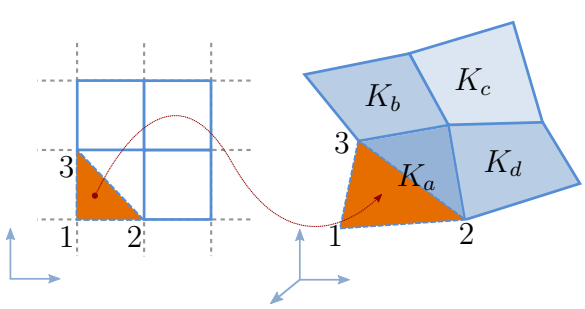
$$E(u,v) = \int_{\Gamma} \Psi_d(|I_1 - I_0^w|^2) + \alpha\Psi_d(|\nabla I_1 - \nabla I_0^w|^2) + \beta\Psi_s(|\nabla u|^2 + |\nabla v|^2) \qquad (5.1)$$

The first two terms in equation (5.1) are commonly referred to as the data terms. They quantify the change in intensities and gradients between the warped source image $I_0^w$ and the target image $I_1$, with $I_0^w = I_0(x+u, y+v)$. The differences in image gradients are a reliable way to match when changes in illumination occur. The

warping helps reduce the displacement within the coarse-to-fine (pyramidal) formulation. The smoothness of the resulting mapping is controlled through the $3D$ gradient. The robust functions $\Psi_d$ and $\Psi_s$ are introduced to alleviate problems related to outliers, noise and occlusions in the data term and, sharpness in the smoothness term. For both cases we use $\Psi(r^2) = \sqrt{(r^2 + \epsilon^2)}$, where $\epsilon$ is a small (in our experiments $\epsilon = 10^{-4}$) term for stabilizing the function when $r$ gets close to zero.

### 5.2.2 Distortion driven variational matching

Commonly, the robust regularizer is used to avoid smoothing across discontinuities which can range from small details to large surface jumps. While it definitely helps curb down smoothing effects at discontinuities, it falls short in many cases, e.g. the bleeding problem in figure (5-3). To address this shortcoming, we introduce $3D$ geometric information to steer grid lines away from largely sheared locations towards more meaningful regions. In this manner, the bleeding problem is addressed indirectly by means of a simple geometric characterization, without requiring any intricate problem reformulation. The idea is to sequentially measure the distortion of the $3D$ reconstructed grid cells and then re-inject these measurements to drive the matching computation. The aforementioned grid is inherited from the image itself, where a cell is given for each pixel in the image. Such grid can be directly projected to $3D$ space since each $2D$ vertex will have an estimated $3D$ location.



For this purpose, we need first to define a geometric characterization of the distortion of each quadrilateral cell. Since triangles represent the simplest element for describing a map distortion, we decompose the quadrilateral cell into two triangles as shown the opposite illustration. We found in our experiments that the condition number $c$ of the jacobian matrix $\mathbf{J}$ of the transform which maps the planar triangle to its $3D$ counterpart works well in general. Figure (5-3) illustrates how this measure clearly captures the regions of interest, in this case, regions with low visibility and relatively high depth variation (shown in blue). In the current setting, the condition number of the jacobian of the transformation can be explicitly expressed according to [57] as $c = c(\mathbf{J}) = (l_{12}^2 + l_{13}^2)/2A$, where $l_{12}$ and $l_{13}$ are the lengths of the reconstructed edges $\{1, 2\}$ and $\{1, 3\}$ resp. and $A$ is the reconstructed triangle area (see figure above). We define the distortion measure $k$ for a cell as the sum of the distortion of its two sub-triangles weighted by their respective inverse areas. That is $k = \frac{c_1}{A_1} + \frac{c_2}{A_2}$.

In order to drive the optimization process, we propose to replace the standard Laplacian operator $div(grad)$ within the robust function in the smoothing term by an anisotropic operator, namely the more general quasi-harmonic operator $div(k\ grad)$. The discretization of the new operator remains similar to the discretization of the smoothing term in equation (5.1), but with small modifications. For instance, for the central vertex in the figure above, the discrete contribution of $u$ in the regularization term becomes $-\beta(r_S u_{y_S} - r_N u_{y_N} + r_E u_{x_E} - r_W u_{x_W})$; where the weights $r_N, r_S, r_E, r_W$ are associated to the derivatives at each side of the vertices in the classical way, see e.g.,[107]. In detail, $r_S$ is given by $r_S = (k_a + k_b)\Psi'_S$, with $\Psi'_S = \Psi'_S(x, y) = (\Psi'(x, y) + \Psi'(x, y+1))/2$. The terms $r_N$, $r_E$ and $r_W$ are defined in a similar manner. The contribution of $v$ for the same term can be derived in the same way as for $u$. In order to maintain a close check on problematic regions the modified flow formulation needs to intervene at each level of the coarse to fine optimization. So at each level of the pyramid we compute the classical solution, measure the induced $3D$ distortion and

then re-inject it into the modified anisotropic formulation. The resulting solution is then transferred to the next level. This leads to the variational matching algorithm 1.

```
/* Start at the coarsest level of the pyramid with w_g ← 0        */
while finest level of the pyramid is not yet solved do
    1. Using w_g as initial state, find a solution w_o using equation 5.1;
    2. Compute the 3D positions for the current solution;
    3. Obtain the values of k;
    4. Obtain w_g using the variational matching with the distortion driven
    smoothing operator (using w_o as initial state);
    5. Move to a finer level in the pyramid;
end
```

**Algorithm 1:** Coarse to fine variational matching of images $I_1$ and $I_2$ using their corresponding projection matrices.

### 5.2.3 Results using distortion driven variational matching

A typical result using the proposed algorithm is shown in figure (5-3-right). The effect of the bleeding problem is clearly reduced to a thin area around the surface jump as substantiated by the projected quality of the triangles (figure 5-3-top-right), and the crisper fountain rim (figure 5-3-bottom-right).

If we observe closeups to same results (see figure 5-4) we notice that points which appear in the rim of the fountain in $2D$, remain in the rim of the fountain in $3D$ when reconstructed using our method. On the other hand, corresponding points will result smeared across the depth jump for the case reconstructed using the method of [18] (see 5-4 to the right). This result exemplifies the correlation between regular triangles and correct $3D$ reconstructions. Further results and evaluations will be shown in section 5.4.

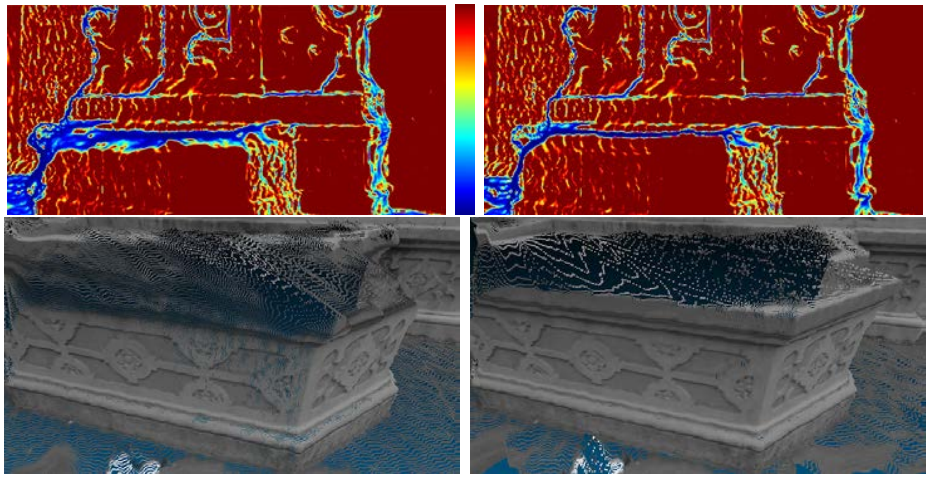


Figure 5-3: Color coded visualization of the quality of reconstructed triangles (projected into the image plane), using the classical formulation (top-left), and our proposed method (top-right). Red and blue represent best and worst quality resp. Close-ups to reconstructions of the fountain's base are shown at the bottom. Please note the crisper rim to the right.

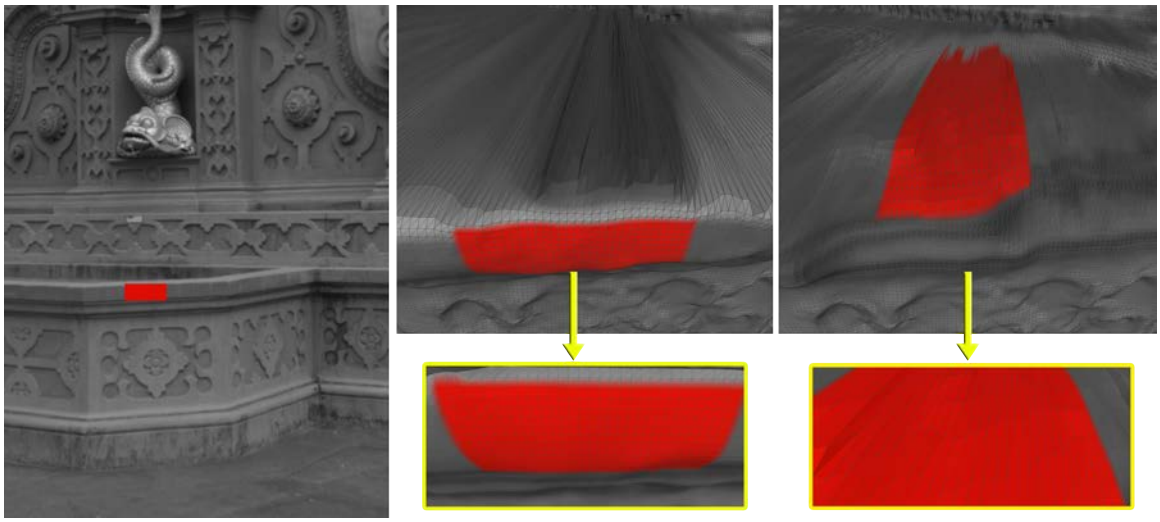


Figure 5-4: Highlighted region on the rim of the fountain (left) and closeups on the results using our method (center) and using Brox et al. [18](right). Our results show a correctly reconstructed rim which displays regular triangles.

## 5.3 Distortion driven multiple-view merging

Our goal, when merging the matches from multiple views, is to select what is best viewed in a given subset of images (triplets and pairs). For this purpose, we rely on Lambert's cosine law which indicates that the image area of a captured scene region increases when the line of sight is closer to a perpendicular configuration. This is illustrated in figure 5-5 above for the cases of flat surfaces (left), smooth surfaces with visible curvature change (middle), and for non-smooth surfaces (right). For the last two cases this area variation becomes more pronounced.

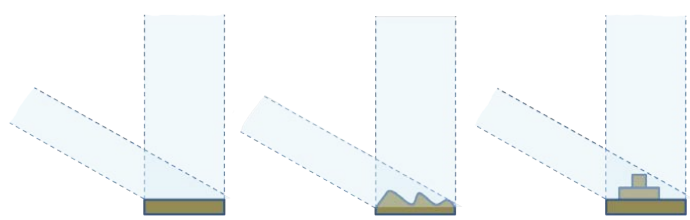


Figure 5-5: Illustration of the area change across two different views in the case of a flat surface (left). This effect becomes more pronounced for smooth surfaces with visible curvature change (middle), and non-smooth surfaces (right).

In practice, we look at the induced distortion when mapping across different views. A simple measure is the local change of the signed triangle area (a negative area tells when the mapping exhibits triangle flips). While this can be a sufficient measure in many cases, it fails to capture deformations which do not affect triangle areas (authalic maps). In general, a planar transformation $S(\mathbf{p}(x,y)) = \mathbf{q}$ which maps a triangle $\{\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}\}$ onto $\{\mathbf{q_1}, \mathbf{q_2}, \mathbf{q_3}\}$ can be characterized by its Jacobian $\mathbf{J} = \left(\frac{\partial S}{\partial x}, \frac{\partial S}{\partial y}\right)$, where the partial derivatives are obtained by means of the divergence theorem:

$$\frac{\partial S}{\partial x} = \frac{1}{A}\int_A (\frac{\partial S}{\partial x}) dA \simeq \frac{\mathbf{q}_1(y_3 - y_2) + \mathbf{q}_2(y_1 - y_3) + \mathbf{q}_3(y_2 - y_1)}{A}; \tag{5.2}$$

where $A$ is the triangle area and $\frac{\partial S}{\partial y}$ can be obtained similarly.

One way to characterize the deformation of the triangle is by looking at the square

roots of the eigenvalues $\lambda_1$ and $\lambda_2$ of $T = J^T J$, which describe the well known elongation along the principal axis when mapping a unit circle onto an ellipse, see illustration in figure 5-6. This eigen-ratio ($\sqrt{\lambda_1/\lambda_2}$) indicates the amount of induced shearing. Combining this measurement with the signed area, we can characterize the overall distortion of the triangles.

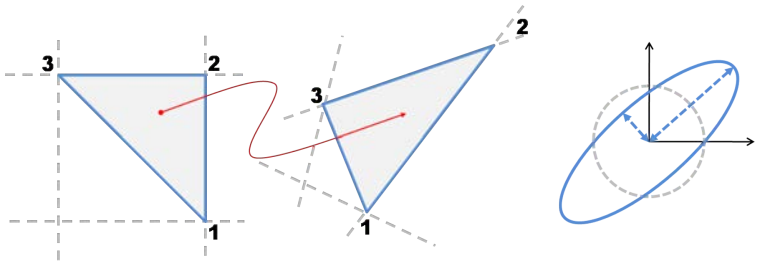



Figure 5-6: Mapping of a (triangulated) regular grid (left) into a deformed configuration. Visualization of the deformation by means of its action on a circle (right). The elongation along the principal axis represents the square roots of the eigenvalues of the transformation.

Using the characterization just described and standard tools from multi-view geometry [46], our merging approach proceeds in two stages. In a first stage, we extract matches which are best viewed in triplets of neighboring views. In a second stage we use best viewed matches in pairs of views to recover regions which were not reliably captured by triplets. For the sake of clarity, let us consider the following scenario, where cameras are ordered in a daisy chain fashion, see figure (5-7). This setting is not a restrictive as it can always be arranged for by measuring the angle between the cameras' principal directions [125].

### 5.3.1 Triplet contributions

Without loss of generality, let us consider an image triplet $\{I_b, I_c, I_d\}$ and its direct neighbors $I_a$ and $I_e$, see figure (5-7-top). First, we compute the dense variational

matching from $I_c$ to $I_b$ and from $I_c$ to $I_d$. In both computations, the grid corresponding the middle image $I_c$ remains fixed. This comes in handy as it yields a direct correspondence between the the three images. Next, the matches $\{I_c, I_b\}$ are transferred to $I_d$ using the trifocal tensor [46] and then the distances between the transferred points and the ones obtained directly from the matching $\{I_c, I_d\}$ are measured. As the computation is not symmetric, we perform the same operation in the other direction as shown in figure (5-7-top). Only matches with an error below a threshold $\epsilon_{trif}$ are kept. In all our experiments this parameter was set to 1. We perform this operation on all triplets (defined by neighboring views) in the image set.

This validation can yield very large data sets. For instance the fountain data set [120] results in more than 60M points. To obtain reliable information out of such large number of points, we observe that errors in the matching manifest themselves as overlaying and interlaying surface sheets, as shown in figure (5-2). The offset between surface sheets can be very tight at places and disturbingly large at others. In order to address this problem, we take advantage of the distortion measure we described previously. We mark a triangle as best viewed in an image triplet if its area decreases and its eigen-ratio deteriorates in neighboring views. To do so, we transfer the matches $\{I_c, I_b\}$ and $\{I_c, I_d\}$ to neighboring images $I_a$ and $I_e$ resp. as shown in figure (5-7-bottom). By inspecting triangles whose areas decay, we can identify which triangles are better viewed in the triplet $\{I_b, I_c, I_d\}$.

Additionally, triangles which pass the area test, need also to satisfy the low distortion measure. We impose $\sqrt{\lambda_1/\lambda_2} > 1/r_{eig}$ and $\sqrt{\lambda_1/\lambda_2} < r_{eig}$ as indicators for acceptable distortion and proceed similarly to the signed area case. In all our experiments, $r_{eig}$ was set to 2 (the optimal eigenvalue ratio is 1).

Lastly, we need to account for points which disappear outside of the triplet as we move to neighboring views. This can be done by simply checking for points which get out the image range when transferred to $I_a$ and $I_e$ or triangles whose areas get close to zero. Such regions are marked in yellow figure (5-9).

It could be argued that the trifocal constraints can be directly incorporated within the matching formulation, e.g. [105]. Although this can improve the results in some regions, it turns out to be more problematic for regions seen only by two views. Because of this, a lot of information is not recovered at particularly interesting regions and, this shortcoming becomes stronger as the baseline increases.

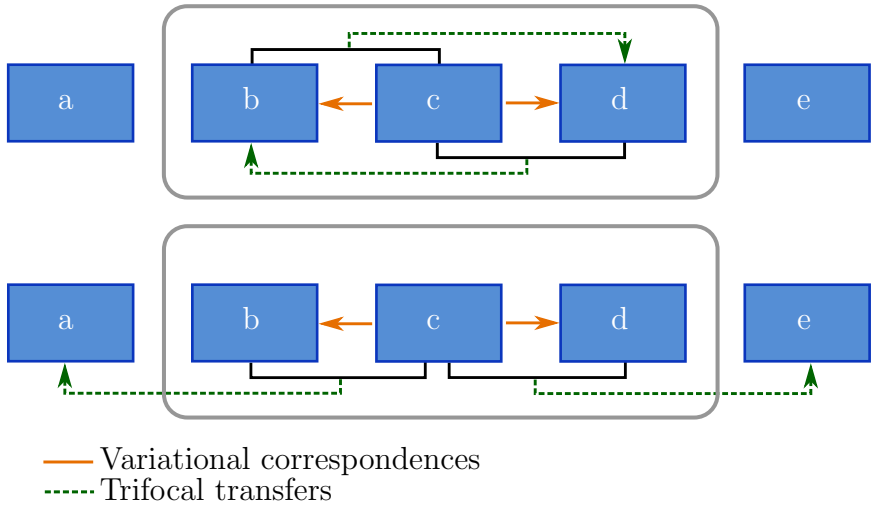



Figure 5-7: Triplet matches are validated based on trifocal transfer within the triplet (top). The selection of best viewed regions in a triplet is performed by transferring the grid to neighboring views and analyzing its distortion (bottom).

### 5.3.2 Pair contribution

Certainly, all points are not necessarily visible in triplets or do not score high enough during triplet validation. Therefore, we still need to account for points visible mainly in pairs of views so as not to miss some important features. In a similar manner to triplets, we need to validate the matches. We have two measures at hand. The first is the distance to the epipolar lines, and the second is by means of the forward backward map [3].

In order to handle visibility for a pair of images, we keep only triangles which

exhibit an area decrease and satisfy the eigen-ratio criteria when transfered to neighboring views. Furthermore, we include points which disappear and triangles whose areas get close to zero. Despite these checks, some outlier triangles can pass all the filters. A simple but reliable way for sorting out these cases, is to transfer the triangles to a slightly faraway view and then measure their distortion, e.g., the pair $I_a, I_b$ can be transferred to $I_e$.

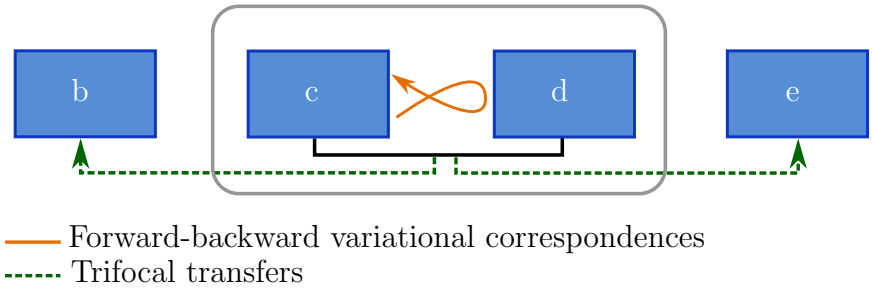



Figure 5-8: Pair matches are validated by epipolar distance and a forward-backward map. The selection of best viewed regions in a pair is performed by transferring the grid to neighboring views and analyzing its distortion (bottom). Only regions which where not covered by triplets are considered.

## 5.4 Results and discussion

We tested our method on several data sets comprising standard benchmarks as well as in-house acquired data. For the reconstruction, either the original pointclouds are shown or the textured reconstructions using [55]. Typical results of our approach are shown for the full sized Fountain-P11 (figure 5-10), and for the Herz-Jesu-P8 data sets (figure 5-11). Both data sets are available from [120]. We benchmarked our results following the method detailed in [120]. In figures (5-10) and (5-11) we show our results obtained and their corresponding error distribution histograms. In the following table we compare the level of completeness and relative errors of our results with those of Furukawa and Ponce [34] and Keriven et al [48].

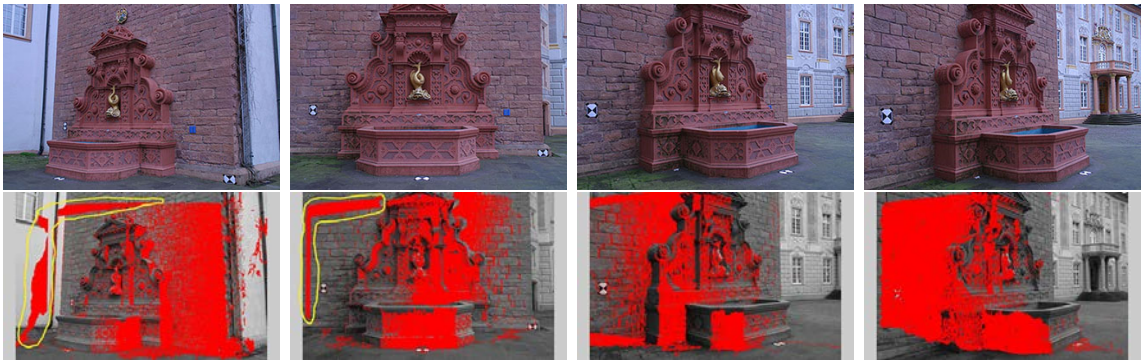


Figure 5-9: Processing best viewed regions in the Fountain data set [120]. Each view represents the central image of a triplet (other two images not shown). The red-colored regions represent areas best viewed in the triplet. Yellow regions represent regions that are only visible in the triplet and therefore are included even if they do not comply with the best view requirement.

| | Fountain | | | Herz-Jesu | | |
|---|---|---|---|---|---|---|
| | Ours | [34] | [48] | Ours | [34] | [48] |
| Relative error | 1.76 | 2.04 | NA | 2.11 | 2.98 | NA |
| Completeness (%) | 85.0 | 79.6 | 90.8 | 81.8 | 80.4 | 91.1 |

For the Fountain-P11 the variational matching computation requires about 10 minutes for a pair. For the same example, which contains eleven $3,072$x$2,048$ images, the merging of the correspondences required only 215s in a Matlab implementation.

We also tested our approach on our own datasets. Results on a face dataset composed of six 1.3 mega-pixels images are shown in figure (5-12). Our approach clearly fares better than the classical formulation, especially at problematic regions such as the hair, ears, chin and neck. Figure (5-13) shows our results on a different face dataset comprising nine 8 mega-pixels images. On this example, our method does a particularly better job reconstructing important features such as the nose and the ears. Lastly, in figure (5-14) we show results for the reconstruction of a boot. This dataset consists of five 4 mega-pixels images. We can observe the chipped boot collar when our distortion driven matching method is not used.

As our approach operates on local neighboring views, its memory requirements are low. In fact, we do not require loading all the $3D$ data at once for processing as it's generally done in related work [34, 48]. In all our experiments, we observed that our merging strategy allows reducing the raw variational matching data by up to a 75%.

We are aware that the proposed merging method is sub-optimal, however we do not see this as a limitation. The locality and low computational requirements of this approach allows reducing the data at hand significantly at a fraction of the computational cost of methods based on global optimization (e.g. [48, 144]). In particular, the results of our approach can be directly used in [48].

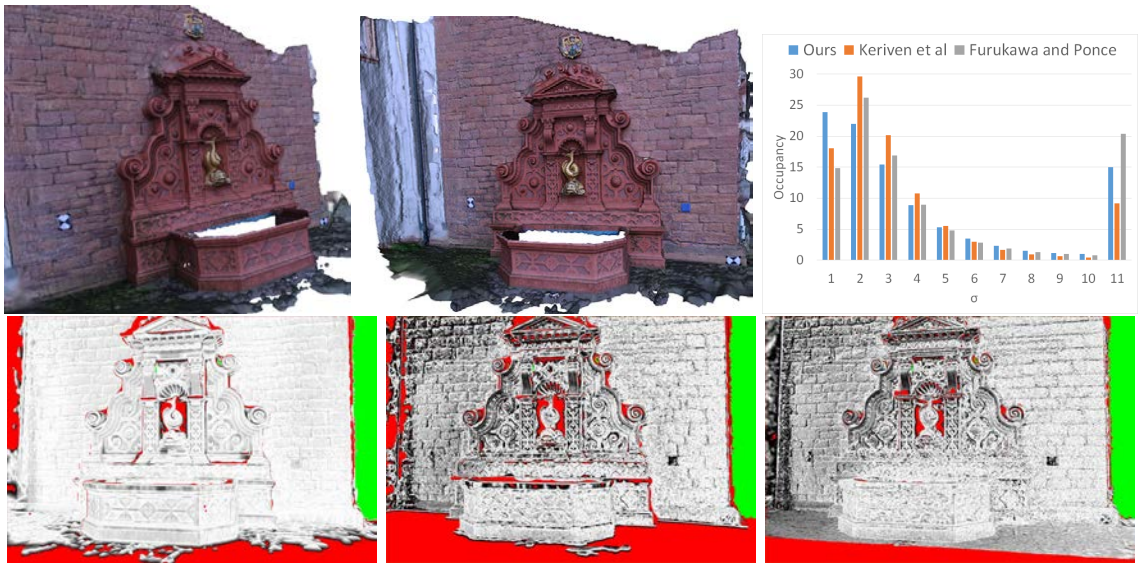



Figure 5-10: Top row, result of our approach on the Fountain-P11 data set [120] and a histogram of the error accumulated for all the views obtained with our method, [34] and [48]. Bottom, distance error for the depth of view five: ours, [34] and [48]. The red color represents no result or result farther than $30\sigma$, green represents the regions where no result can be obtained, dark gray represents larger error (smaller than $30\sigma$).

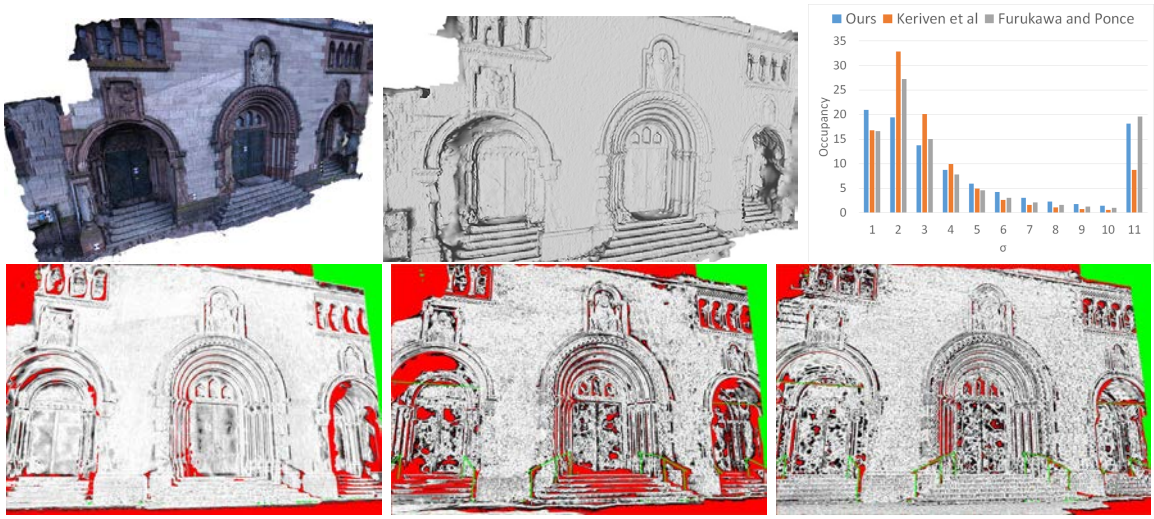



Figure 5-11: Top row, result of our approach on the Herz-Jesu-P8 data set [120] and a histogram of the error accumulated for all the views obtained with our method, [34] and [48]. Bottom, distance error for the depth results of view five: ours, [34] and [48]. The red color represents no result or result farther than $30\sigma$, green represents the regions where no result can be obtained, dark gray represents larger error (smaller than $30\sigma$).

## 5.5 Conclusion

In this chapter, we proposed two systematic enhancements to the classical variational scene reconstruction. For the variational matching, an adaptive approach allows recapturing lost details by means of an anisotropic diffusion driven by geometric distortion. At the merging level, a selective technique for obtaining the best contributions across neighboring views allows reducing data redundancy. Unlike most of related work, the approach resolves a large number of outliers cases prior to estimating the final $3D$ point cloud.

All of our contributions are achieved by means of a simple, yet principled, characterization of geometric deformations and, can be easily reused in other methods within the variational context. Our approach is fairly straightforward, simple to implement,

and our results can be easily reproduced.

In future work we can explore the idea of having adaptive thresholding values for the merging stage described in section 5.3 since they are currently defined globally.

Figure 5-12: Two images from face2 dataset-6 $\times$ 1.3 MP- (top) reconstructed using our approach (middle) and using variational matching code provided by the authors of [18].

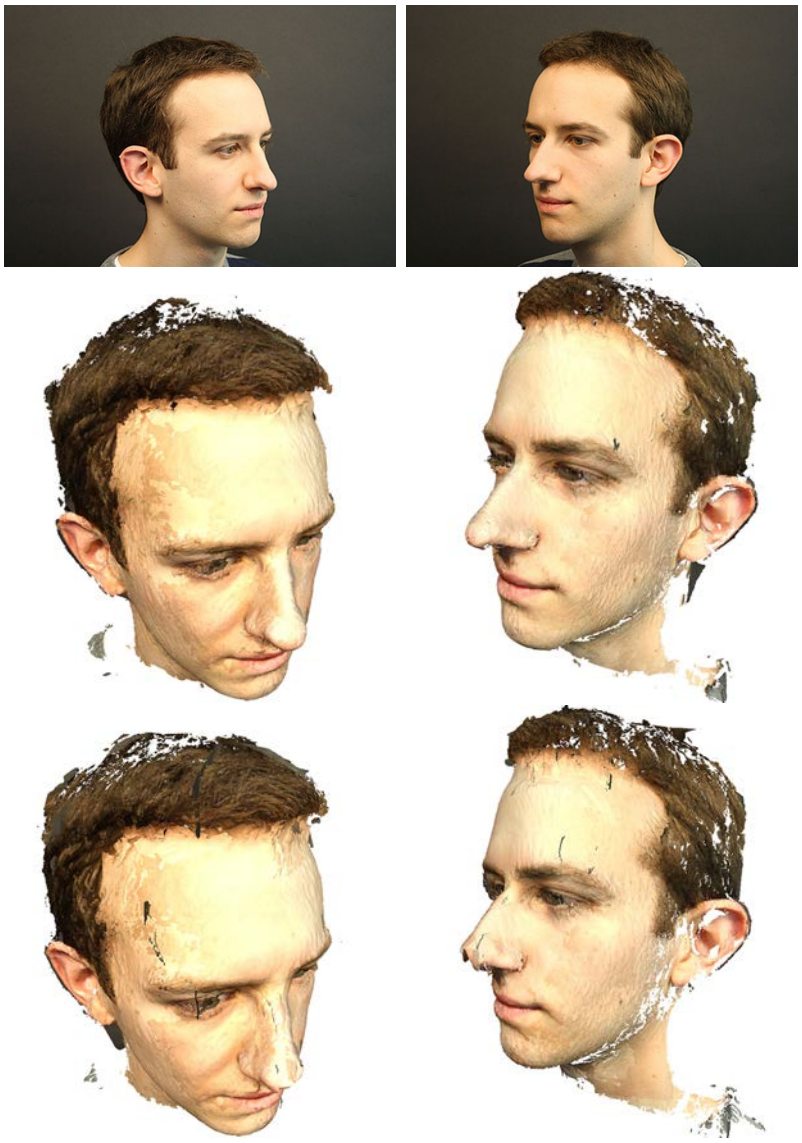


Figure 5-13: Two images from face1 dataset-$9 \times 8$ MP- (top) reconstructed using our approach (middle) and without the use of our distortion driven matching.

Figure 5-14: Two images from our boot dataset-5 $\times$ 4 MP- (left) reconstructed using our approach (middle) and without the use of our distortion driven matching (right).

# Chapter 6

# Propagation-based matching for 3D reconstruction

This chapter describes the evolution of propagation-based-matching for 3D reconstruction algorithms. Starting from early work that developed the concept to be used with satellite images finishing with the state of the art methods developed in the last decade, which provide the means to find matches in wide baseline conditions. This chapter intents to depict the evolution of the concept but it is not meant to represent a full literature review on it.

## 6.1 Introduction

The propagation and region growing techniques mainly stem from classical segmentation approaches [44, 85]. In the segmentation context, the idea consists in merging neighbors that share common properties, into larger and homogeneous regions. In the case of 2D intensity-based image segmentation (e.g. common pictures), the neighbors can consist of adjacent pixels that can be merged if, for example, they present a similar intensity value. In the context of matching, propagation works by first inheriting

information that can be of help in the correspondence search. Once a correspondence is found, local information is updated to produce more accurate matching and more relevant information to propagate locally.

Match propagation consists in obtaining new matching points around already matched locations. In this manner a reliable sparse set of matches can be propagated to a quasi-dense map between two views $(I_1, I_2)$. This approach starts by finding initial matchings $S = \{\mathbf{s}(\mathbf{x_1}, \mathbf{x_2}), \mathbf{x_1} \in \mathbf{I_1},\ \mathbf{x_2} \in \mathbf{I_2}\}$, called seeds, which are commonly obtained by feature detection and matching techniques (see Chapter 3). The propagation follows by cyclicly expanding the correspondences in the neighborhood of the seeds . Each correspondence found during the propagation steps is considered a new seed that can be, at some point, propagated. Ideally, at least one initial seed should be found for each isolated image region. An isolated region is defined as a region that cannot propagate into any other region, either because it is surrounded by an un-textured belt or because it represents a spacial surface patch disconnected from its surroundings.

Matching propagation and region growing algorithms come a long way in the literature and find its roots in the work of Otto and Chau [97], where the authors propose a matching algorithm for the 3D reconstruction from images taken from satellites. In their method the initial seeds are determined by hand and each neighbourhood is defined by deforming and matching a patch around the seeds following the least squares correlation approach by Gruen [40]. This approach aims to approximate the local deformation by an affine transformation. The six parameters of the affine transformation are estimated by minimizing the sum of the squares of the differences between a patch in $T_1$ and a transformed patch from $I_2$. For this minimization to converge, the corresponding pixels inside the patches cannot be farther away than 1-2 pixels for the minimization to converge. After each optimization is done, and if it converges, the resulting correspondences are marked as matches and used as new seeds.

There are several shortcomings to this method, including the fact that this optimization process is time consuming. It is also important to note that in this approach there is no matching validation, which can lead to severe mismatches. Despite of these drawbacks, the work presented in [97] laid down the main ideas for the matching propagation methods that followed.

The improved propagation algorithm presented in [64, 65] starts by automatically obtaining a set of sparse matches [112, 45], and propagates neighboring pixels that belong to image patches that have a correlation score larger than a threshold. The correlation score used is the zero-mean normalized cross-correlation (ZNCC) denoted in equation 6.1. This correlation score is used to match the initial set of sparse features detected and to match corresponding neighbors. The scores obtained are not only important to determine correspondences but also to determine which pair of matches will propagate first, taking locations that cannot be taken again by other matches.

$$ZNCC(I_1(x), I_2(i)) = \frac{\sum_d (I_1(x+d) - \bar{I}_1(x))(I_2(i+d) - \bar{I}_2(i))}{(\sum_d (I_1(x+d) - \bar{I}_1(x))^2 \sum_d (I_2(i+d) - \bar{I}_2(i))^2)^{\frac{1}{2}}} \tag{6.1}$$

where $\bar{I}_1(x)$ and $\bar{I}_2(i)$ are the mean pixel intensities for windows centered around the locations $x$ and $i$ in images $I_1$ and $I_2$ respectively.

## 6.2 Match propagation for wide-baseline configurations

While the method of [64, 65] (described above), can work well for narrow baselines it becomes more and more inaccurate as the baseline between the images under consideration increases, up to a point that regions cannot longer be matched. The reason is that the windows used for cross correlation no longer correspond to the same shape, size and/or orientation in both images. For example, in figure 6-1, approximately

corresponding windows for narrow to wider baselines are illustrated (increasing towards the right and bottom). The use of the rectangular (green) windows centered around corresponding points introduce increasing error to the region matching as the baseline becomes wider. The blue windows adapt to the local transformation so they approximately represent corresponding regions.

To deal with this region matching problem, Ferrari et al. [29, 30] propose not only to allow certain local deformations but to actually account for them as local affine transformations. In [29, 30], the affine invariant interest point detector [80] is used to obtain seed candidates that are matched using the Mahalanobis distance. Their subsequent propagation does not follow a winner takes all principle. Instead, it promotes the competition for propagation among nearby matches. Each propagation step is followed by a topological filter step that removes points based on a sidedness constraint. This filter aims to remove points that switched sides from one image to the other (see figure 6-2).

The matching score ($sim$) is obtained using a combination of normalized cross correlation and an averaged Euclidean distance in RGB color space over the affine-transformed image patches:

$$sim(A(I_1), B(I_2)) = NCC(A(I_1), B(I_2)) + (1 - \frac{dRGB(A(I_1), B(I_2))}{100}) \qquad (6.2)$$

where $dRGB$ represent the average per-pixel Euclidean distance after independent normalization in the three channels (done to gain photometric invariance); $A(I_1)$ represent affine-transformed image patches of $I_1$ and similar the same goes for $B(I_2)$. The normalized cross correlation (NCC) is computed over grey-scaled versions of the affine-transformed image patches.

After a new matching point is found [30] proposes to refine the affine transformation by searching in a bounded affine space, searching for the six parameters (translation $(u, v)$, rotation $\theta$, shear $(d_x, d_y)$ and scale $s$) that produce the highest correlation

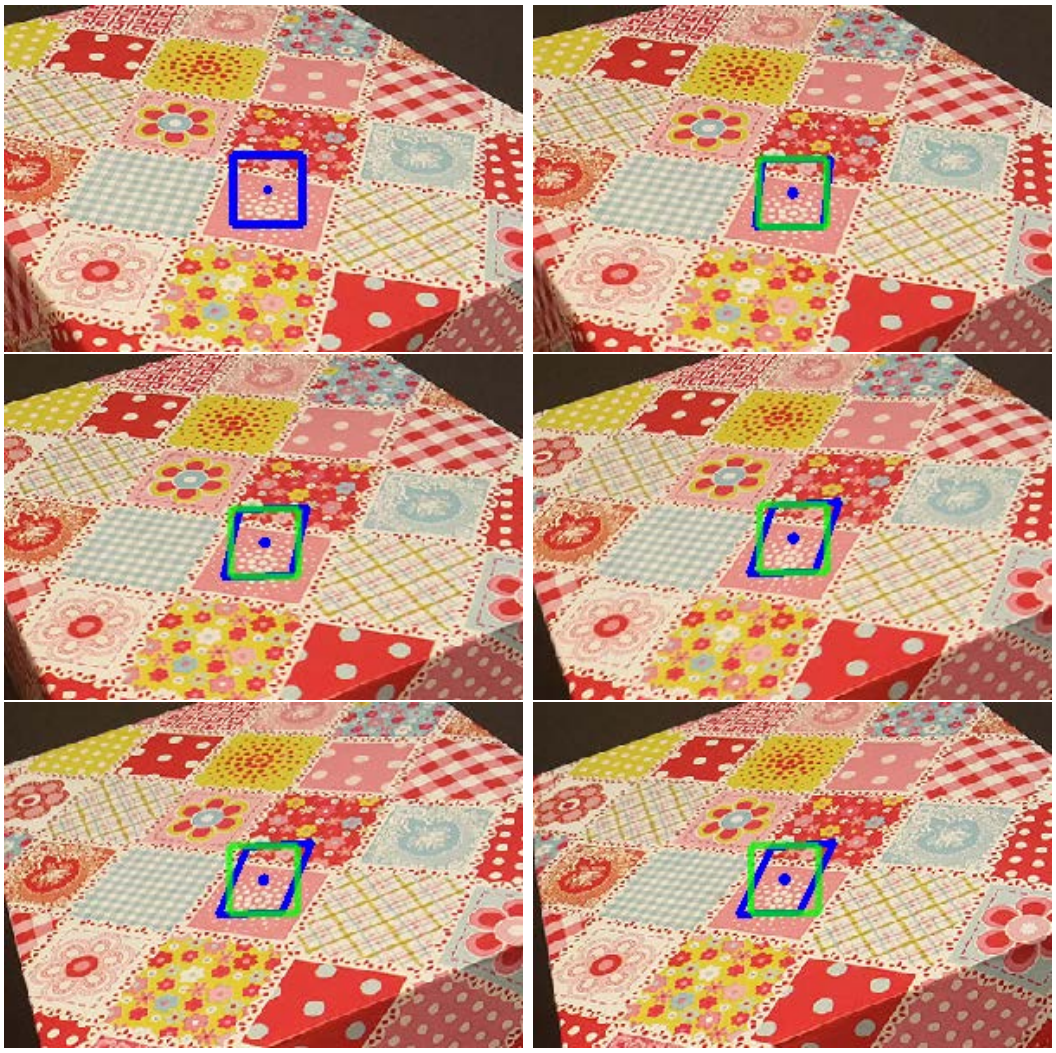


Figure 6-1: Window correspondences from narrow to wide baselines (going from left to right and top to bottom). This figure illustrates how rectangular windows (in green) around corresponding points, do not faithfully represent corresponding regions. On the other hand, windows adapted through an affine transformation (in blue) clearly approximate better the surface at hand.

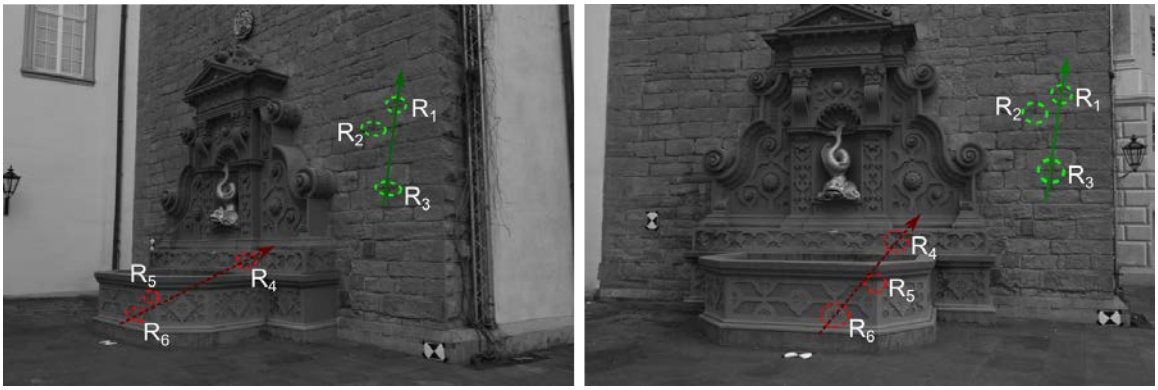



Figure 6-2: Illustration of the sidedness verification across two views. Regions $R_1, R_2, R_3$ pass the verification since, in both images, $R_2$ belongs to the left side of the vector that connects $R_3$ to $R_1$. For the case of the regions $R_4, R_5, R_6$ the sidedness check fails since $R_5$ changes sides across the vector that connects $R_6$ to $R_4$.

between corresponding patches in two images.

The method proposed in [30] does not aim to produce a number of matches that can be triangulated to represent scenes or objects; its purpose is to simply provide enough region matches, across multiple views, to perform object recognition. Nevertheless, it lays down five important ideas in the context of image matching. i) The seed region extraction and matching can be efficiently performed in an automatic manner, without the need for human interaction. ii) Correlation scores that include normalization are important to confirm matches in images obtained in uncontrolled illumination conditions. iii) Local affine transformation estimations are of great importance to achieve accurate and more complete matching for cases of wide-baseline. iv) Local affine transformation estimations can be passed on to candidate matching neighbors. This works particularly well for neighboring regions that belong to the same plane in the scene but it also provides a good initial estimate for regions that belong to smoothly varying surfaces. v). Finally, the inherited local affine transformations should be updated to increase matching accuracy and to extend the life of

the propagation line.

In the next section we outline the method of Kannala and Brandt [54], which uses local affine estimations to obtain quasi-dense matches in wide-baseline setups.

## 6.3 Quasi-dense match propagation for wide-baseline configurations

In this section, we briefly summarize the key ideas in the quasi-dense propagation work presented in [54]. The main steps presented here are similar to those already presented in this chapter: affine feature matching followed by an affine match propagation which updates the transformation estimations. In more detail, the region detection used is [81] and the descriptors for matching is the one in [74]. Once the initial feature matches (seeds) are obtained and stored in a priority queue (along with their corresponding affine transformation estimations $A$), the propagation proceeds by repeating the following steps:

1. remove the seed $s$ with best score from queue
2. search new candidates in a neighborhood of $s$ based on local affine transform $A$
3. compute correlation scores for all candidates
4. append candidates to the matches and seeds if they have high correlation score, satisfy the epipolar constraint and are not yet filled
5. update the candidate's affine transform based on optical information $A \leftarrow A_O$
6. mark the corresponding pixels as filled

The correlation score is measured using the zero-mean normalized correlation (ZNCC) of geometrically normalized image patches along with the minimum intensity variance of the two patches.

When dealing with wide-baseline views, it is useful to have the fundamental matrix between the views as well as the local affine transforms associated with the individual seeds [54]. The fundamental matrix can be used to filter candidate matching neighbors using a small threshold (usually less than two pixels) to gain flexibility against calibration errors and distortion due to image acquisition equipment. The local affine transformation estimation is used, as in previous works [97, 29] to normalize regions so they closely correspond to matching areas. On top of the common latter consideration and in order to reduce geometric distortions, the reference view is defined for each seed so as the transformation between the two views is always magnifying. This means that for some cases the transformed patch will be in $I_1$ and in other cases will be in $I_2$. This simple idea can bring an increment in the accuracy of the matching as shown in figure 6-3, where the $ZNCC$ matching score increases for the case of magnifying affine transformation (patches on the right column) with respect to the case of reducing affine transformation (middle column) from 0.69 to 0.83 (1 being the highest correlation score).

A key contribution of the work presented in [54] is the inexpensive manner in which the affine transformations are updated. This is done through the use of the second order intensity moments and epipolar geometry. Using their affine transformation update, more complete propagations are achieved, covering regions where the local transformations vary significantly from that of the initial seed(s). The main idea, similar to [71, 81], follows the notion that corresponding patches can be represented as $f_1(x) = f_2(Ax)$, where $f_1$ and $f_2$ are patches in $I_1$ and $I_2$, and $A$ is a $2 \times 2$ (local) affine transformation matrix that does not take into account translation. To simplify computations, the inherited scale estimation can be assumed to be close to correct so the local affine transformation can be obtained estimated as $A = S_1'^{-(1/2)} R S_2^{(1/2)}$, where $R$ is a rotation matrix obtained using epipolar information [23] and $S_1$ and $S_2$ are the windowed second order intensity moment matrices of $f_1$ and $f_2$.

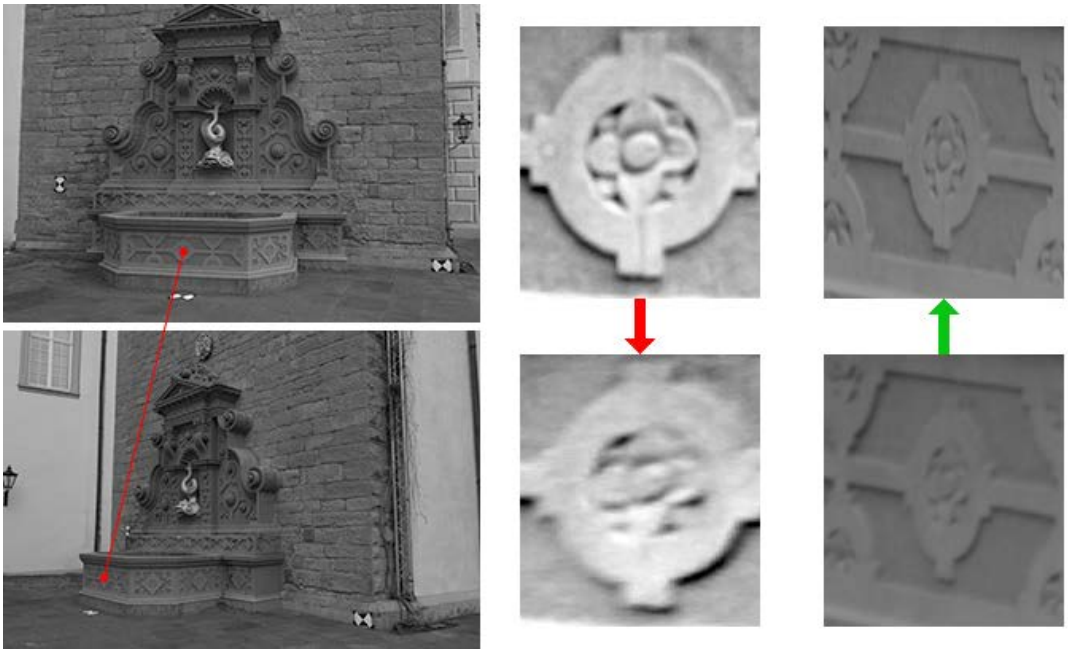


Figure 6-3: Illustration of the effect of choosing a magnifying affine transformation instead of a reducing one. Left column, two views of a scene with matching points denoted by a connecting red line. In the middle column, corresponding windows (around matching points) using a reducing transformation that achieves a correlation score of 0.69. On the right column, corresponding windows (around same matching points as in the previous example) using a magnifying transformation that achieves a correlation score of 0.83

The method just outlined here allows obtaining accurate and quasi-dense matching results at a low computational cost, which directly enables a direct computation of 3D models from images. Examples are shown and discussed in 6.3.3.

### 6.3.1 Quasi-Dense Wide Baseline Matching for Three-Views

An extension of Quasi-Dense Wide Baseline Matching previously presented, was proposed in [61], where a third view is included to improve matching results. The improvements using this extension come as an increased matching accuracy but, most

of all, they are noticeable in the reduction of outliers due to the best-first matching approach used in the presence of repetitive patterns or lowly textured regions. The use of a third view is directly possible by transferring candidate pair-wise matches to a third view using trifocal tensor transfer [46]. In order to obtain the trifocal tensor one needs to either have the projection matrices or compute a (small) set of matches between the tree views and obtain the tensor using an existing method (e.g. the robust six point correspondence method proposed in [126]).

### 6.3.2 Multi-view quasi-dense matching for wide-baseline configurations

The method proposed in [54] and even its extension proposed in [61] remain somehow oblivious to the 3D nature of the scenes (when used in such context). Therefore, [142] proposes to use the latter method in a multi-view context, using a prioritized matching to first expand the most promising correspondences in a scene taking into account multiple views at the same time. It starts by acquiring pairwise affine invariant feature matches [83] and reconstructing their 3D positions (calibrated scene is assumed). The feature matches are stored along with a reference to their corresponding views and a ZNCC score. During each propagation step, the best seed is used to propagate to in the immediate neighbourhood by computing correlation scores and keeping a set of propagation candidates that pass a minimum correlation threshold. The propagation candidates are then triangulated and projected to other views to compute their correlation scores. New matching scores are obtained by combining pairwise correlation scores and minimum intensity variances. Candidates with the best scores are kept and are used as seeds to propagate further. In equation 6.3 we formulate the exact definition of the combined matching scores, where $k$ is the number of views, $z_k$ the pairwise correlation score and, $t$ a threshold value.

$$mvs = \sum_k \max\left(0, 1 - \frac{(z_k - 1)^2}{(t-1)^2}\right) \tag{6.3}$$

This simple and direct application of image-based match propagation to multiple views achieves high quality results that are comparable to those of other state of the art methods (e.g. [34]). The reader can refer to [142] for examples of using this method in the 3D reconstruction of real scenes, along with benchmarks and a comparison to [35].

### 6.3.3 Results and discussion

We compare results obtained using the methods proposed in [54] and [64] in the Fountain and Herz-Jesu scenes proposed in [117]. The results are shown in figures 6-4, 6-5 and 6-6. The performance of the results are computed by measuring the depth error estimations compared to the ground truth. The charts shown in the figures represent the cumulative error compared to the occupancy percentage of the results. Note the higher accuracy and completeness obtained using method by [54], compared to the non-adaptive method of [64]. This higher performance becomes more evident for wider baselines, as seen in figure 6-5, where not only higher accuracy and completeness is obtained, but also, the method by Kannala and Brandt is less affected by the increase in baseline.

Note that the depth error charts presented in figures 6-4 and 6-6 also illustrate results for the method of [34], presented int he next section. This method, despite using multiple views to perform match propagation and reconstruction along with several filters, it does not achieve significantly higher accuracy or completeness of the results. In fact, when we measure the error and occupancy for just two views( 6-4), the method of Kannala and Brandt attains a higher score that of Furukawa and Ponce. It can be argued that Furukawa and Ponce bring other, highly significant,

benefits to the reconstruction problem, such as robustness and the embedded use of multiple images, making these two methods difficult to compare. What we would like to convey here is that, in most cases, more information can be obtained from matching step alone. This last idea is the initial motivation for our propagation based stereo reconstruction method that we will present in chapter 7.

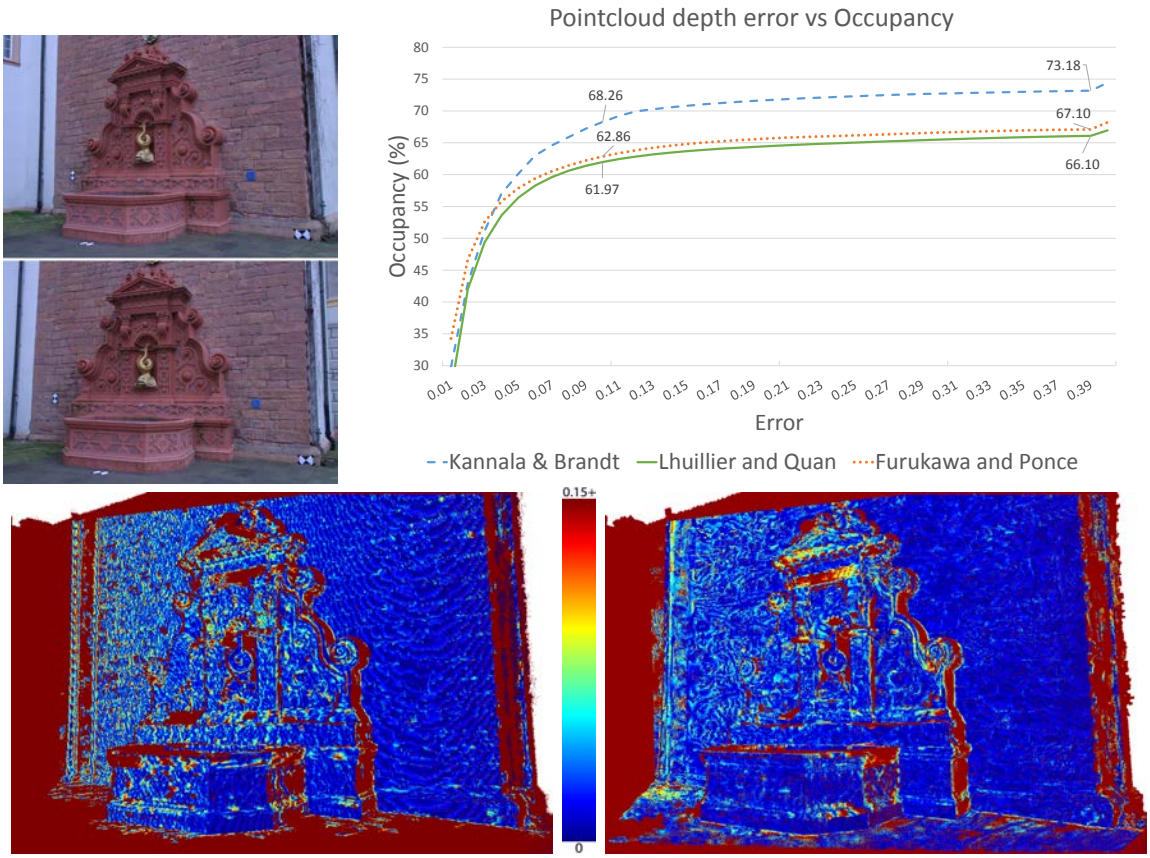



Figure 6-4: Depth error comparison for the Fountain-P11 scene. In the top row, the views two and three used to reconstruct the scene (left) and a cumulative depth error chart for the current views using methods [64], [54] and [34]. In the bottom row, color coded depth error results using methods [64] (left) and [54] (right). Color code bar (center), where dark blue represents errors close to zero and dark red represents errors equal or greater to 0.15.

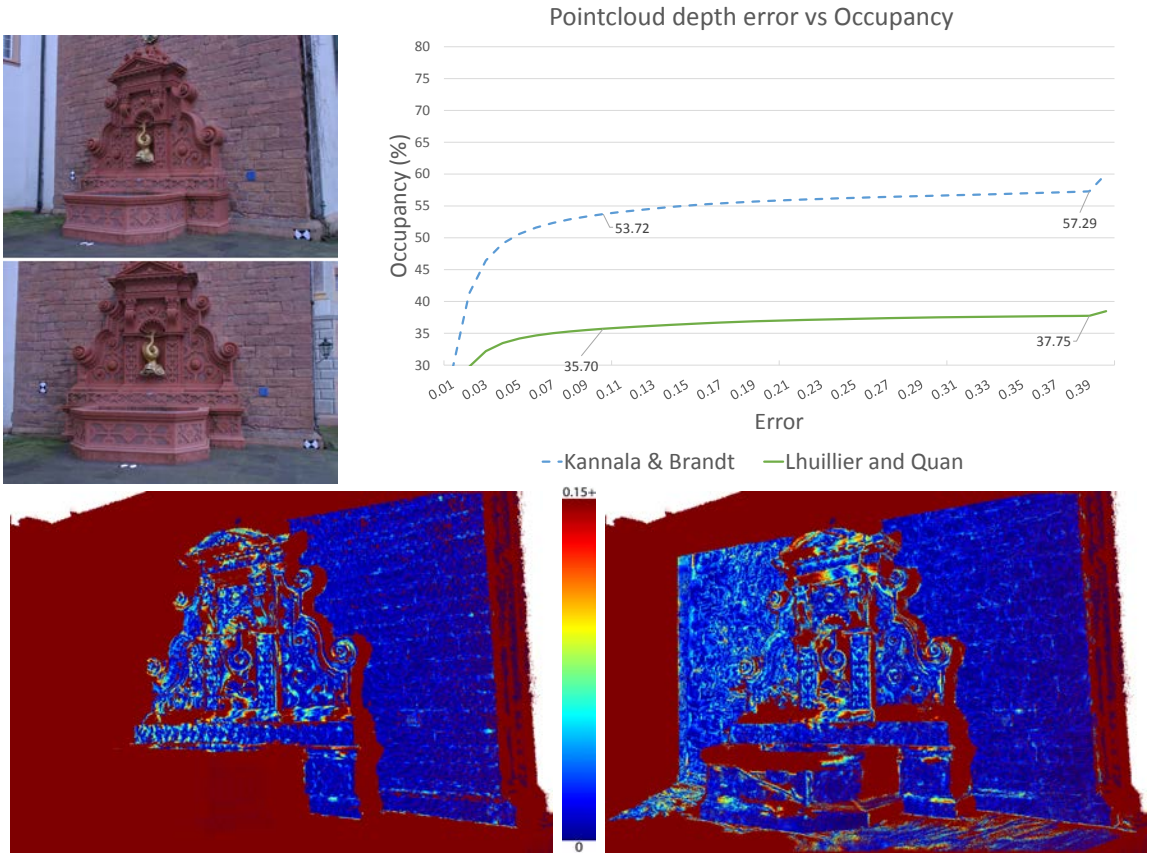



Figure 6-5: Depth error comparison for the Fountain-P11 scene for a case of a wider baseline. In the top row, the views two and four used to reconstruct the scene (left) and a cumulative depth error chart for the current views using methods [64], [54] and [34]. In the bottom row, color coded depth error results using methods [64] (left) and [54] (right). Color code bar (center), where dark blue represents errors close to zero and dark red represents errors equal or greater to 0.15.

## 6.4 Accurate, Dense, and Robust Multiview Stereopsis

In the context of 3D reconstruction from images, [35] presents a technique based on match propagation that uses multiple views. This technique aims first to obtain

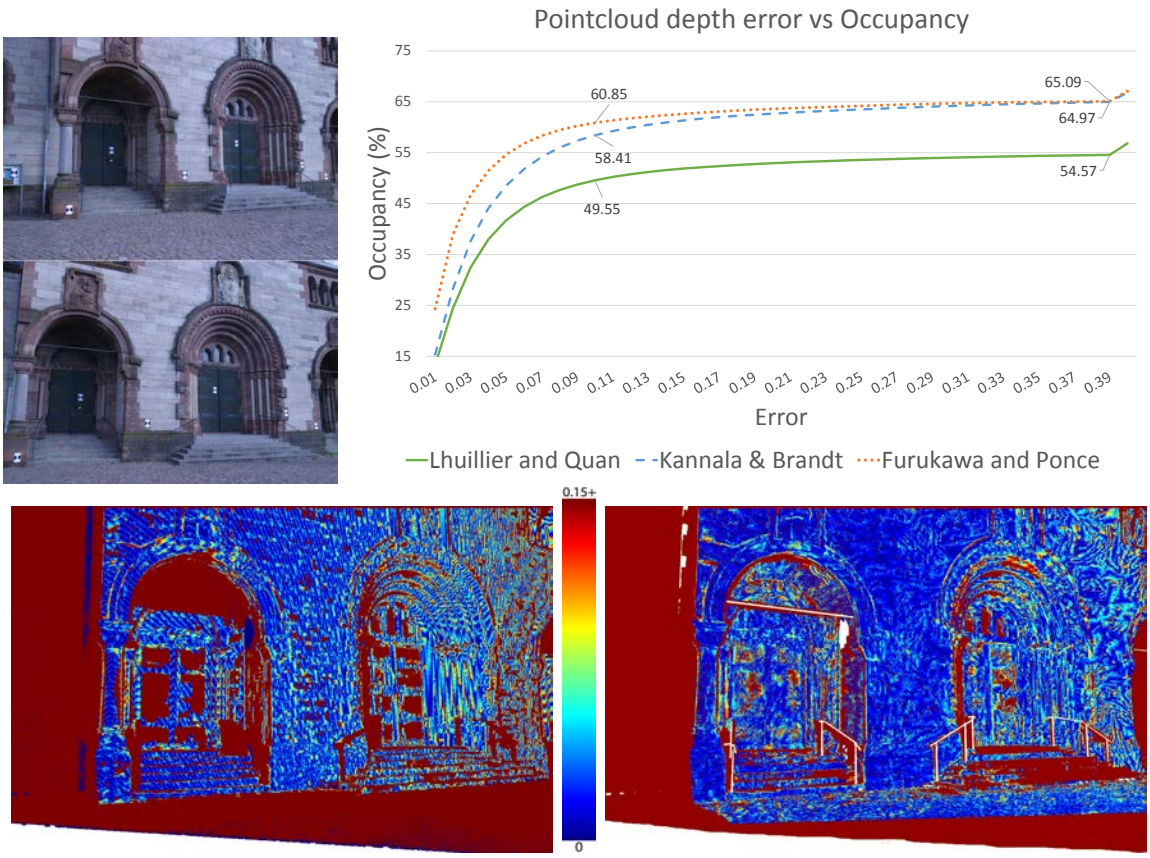



Figure 6-6: Depth error comparison for the Herz-Jesu-P8 scene. In the top row, the views three and four used to reconstruct the scene (left) and a cumulative depth error chart for the current views using methods [64], [54] and [34]. In the bottom row, color coded depth error results using methods [64] (left) and [54] (right). Color code bar (center), where dark blue represents errors close to zero and dark red represents errors equal or greater to 0.15.

spatial oriented points (patches) that are defined by its centers and unit normals and are linked to the views with which they were obtained. It then obtains a polygonal mesh model, initialized using [55] and, refined via an energy minimization that takes into account per vertex photometric discrepancy and geometric smoothness. This mesh refinement will not be explained here since it is not of direct interest in this

chapter, but it is important to note that their match propagation algorithm follows three steps: feature (detection and) matching, propagation and filtering. With the last two (propagation and filtering) repeated a few times (around 3).

For the feature detection Difference-of-Gaussian (DoG) and Harris [45] operators are used to detect blobs and corners. The detected features are matched, pairwise, all against all and filtered. The first filter keeps candidate correspondences that lay close to their corresponding epipolar lines (the value of two pixels is used). Next, the pairs are triangulated and are only kept if the angle formed between the rays that pass through the spatial points and the corresponding optical centers is higher than a threshold. After these two filters are applied, correlation scored for the pairs are computed and kept only if they lay above a threshold. At this stage, chains of matching features that reference at least two views were obtained and are later refined in 3D to achieve better 2D correlation scores.

During the propagation stage, features are attempted to propagate to neighboring cells. A cell represents an area of $n \times n$ pixels and the aim of the algorithm is to obtain one match per cell but, in practice, several matches could live in a single cell. Propagation is done by selecting a chain of correspondences not already propagated, setting neighboring cells as candidate correspondences and refining their positions in the same manner as done for the features. Visibility of the candidates is updated using a depth map computed using cell-wise information. Candidate correspondence chains are validated if they are at least visible in a minimum number of views (3 is the recommended).

Three filters are used to remove erroneous matches obtained in the previous propagation stage. First, two matches that live in the same cell but are not neighbors are detected as problematic and the matches that produce more 3D points are kept (see figure 6-7-left). Next, a point is removed if it is determined to be visible in less than a minimum number of views; with visibility is obtained using a depth map test (see figure 6-7-right). As a last filter, for each point $p$, a set of points $N$ laying in the

same or neighboring cells are obtained and a the point $p$ is kept only if its neighbors represent at least 25% of $N$.

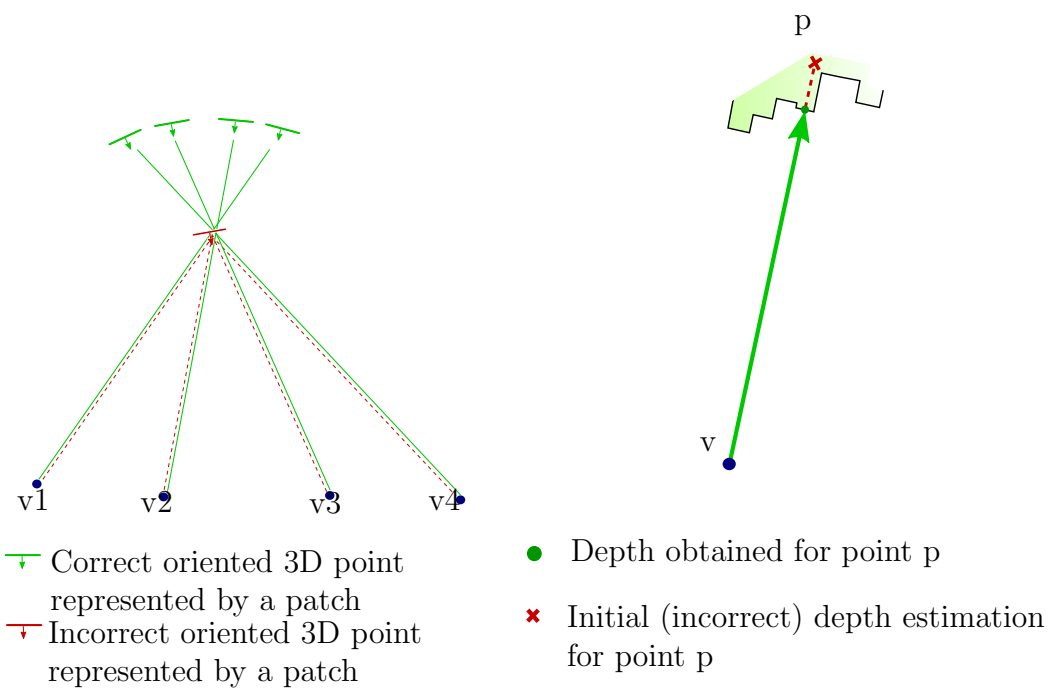



Figure 6-7: Two of the three filters used in [35]. **On the left**, first filter that verifies neighboring matches and keeps points closer to the cameras. **On the right**, a common visibility computation using a depth map test. In this case, the point $p$ marked by a red cross does not pass a visibility test.

### 6.4.1 Results and discussion

The method partially presented in this section has gained great recognition in the subject of 3D reconstruction and it is almost always used for comparisons. It uses main ideas from the propagation based methods previously presented in this chapter, but it recognizes the need for certain regularization that takes into account immediate neighborhood. Although the nature of the propagation has already embedded certain relation or regularity within the neighborhood, it is clear that it is not enough [35]. To

attend this need the author proposes a series of propagation stages that do not make use of neighboring information, followed by fixing stages that partially incorporate neighborhood information into the process. The obtained pointcloud after this stages is still not precise enough, and contains noise and outliers. Therefore, the authors initialize their mesh, using a global optimization solution [55] that does not necessarily pass through the obtained 3D points, loosing much of the information and precision of the data obtained, but solving for the sparseness and outliers problems.

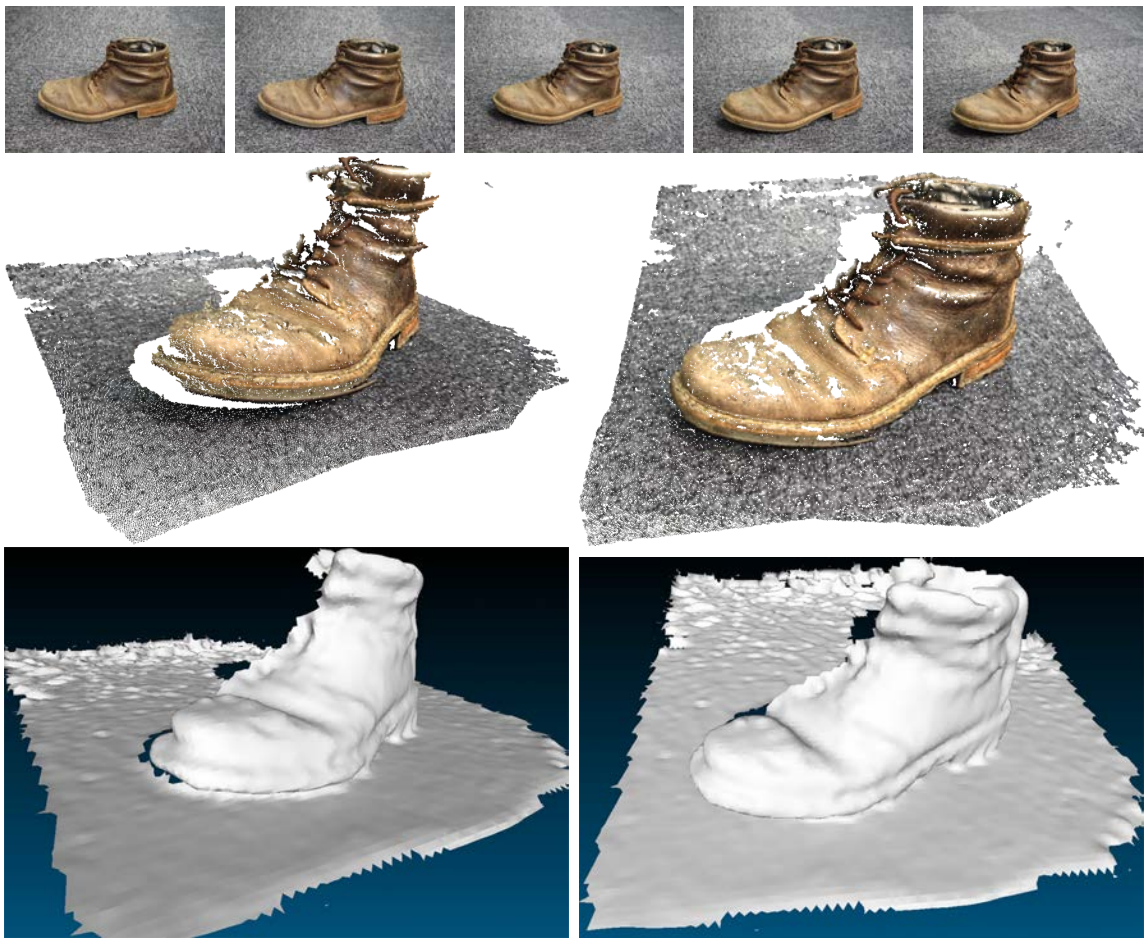


Figure 6-8: Example of a reconstruction using [34]. Top row, four $1,944 \times 1,296$ pixels images used for reconstruction. Second row, two views of a point cloud obtained using [34]. Bottom row, two views of a meshed version of the upper pointcloud using [55]

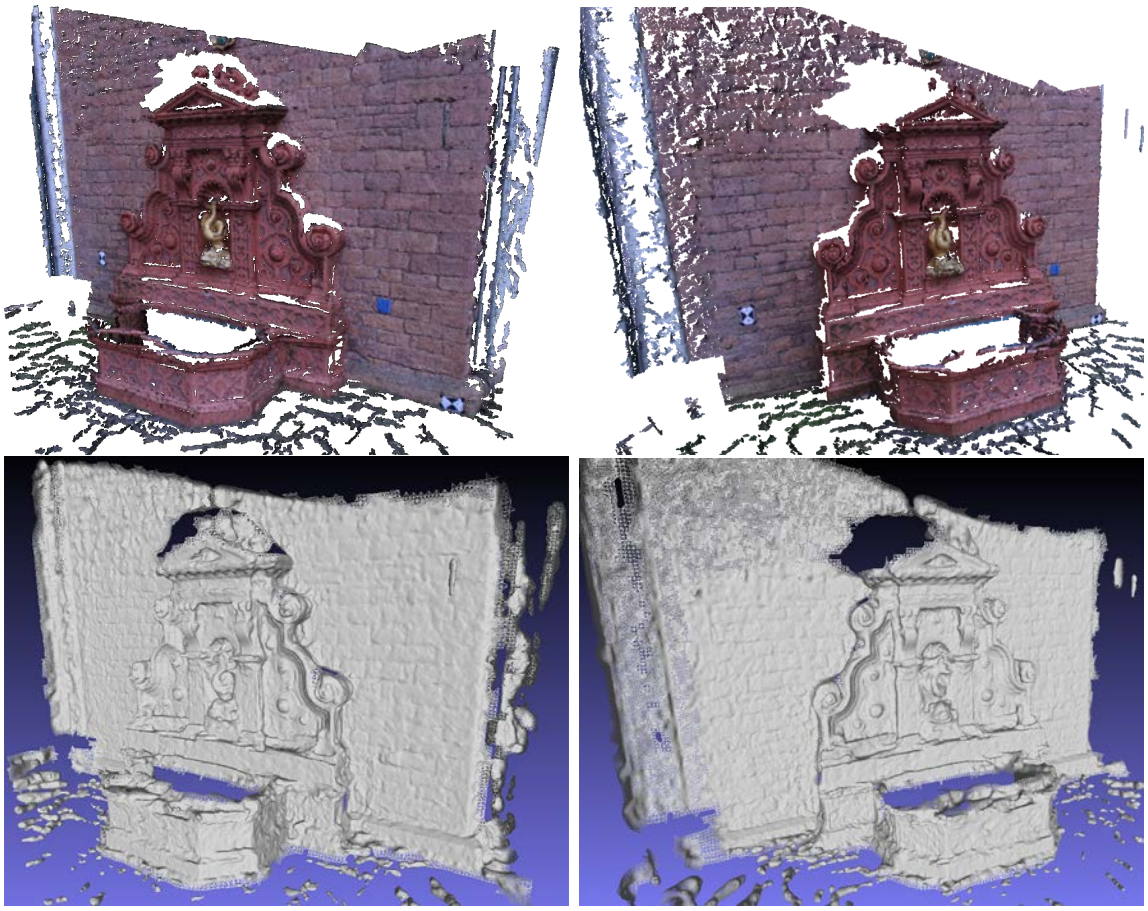


Figure 6-9: Example of a reconstruction using [34] for the Fountain-P11 dataset. Top row, two views of a point cloud obtained using [34]. Bottom row, two views of a meshed version of the upper pointclouds using [55]

# Chapter 7

# Complementary geometric and optical information for match propagation based 3D reconstruction

While in general, propagation based methods capture well the scene structure, the recovered geometry often presents a choppy nature which can be attributed to matching errors and variations in the estimated local affine transformations. In this chapter we propose to control the reconstructed geometry by means of a local patch fitting which corrects both the matching locations and affine transformations throughout the propagation process. In this manner, matchings that propagate from geometrically consolidated locations bring coherence to both positions and affine transformations. Results of our approach are not only more visually appealing but also more accurate and complete as substantiated by results on standard benchmarks.

## 7.1 Introduction

This chapter revisits the problem of match propagation across a pair of views in the context of quasi-dense matching [65], and its extension to wide-baseline views which accounts for affine distortion [54] (presented in the previous chapter). We focus on these approaches due to their inherent simplicity and minimal requirements, however the proposed solutions can be applied to other propagation approaches. Alternative approaches such as [22, 21] which require image rectification and further problem reformulations will not be addressed here.

Propagation-based matching is the workhorse of many surface reconstruction approaches e.g. [66, 141, 34, 61]. In practice, it comes in different flavors, e.g. [65, 97, 30], but they all require corrective steps and/or postprocessing techniques to filter out mismatches. Furthermore, the resulting geometry exhibits a choppy nature which is often not addressed directly and is left to mesh reconstruction algorithms, e.g. [55]. Figures 7-1 and 7-2 show typical chaps and clefts observed when performing scene reconstruction using a state of the art propagation approach, in this case [54]. We are aware that the use of additional images can help reduce such effects relatively, e.g. [142, 34], however, addressing the challenges of the fundamental problem on a pair of images remains necessary to take full advantage of the available data and can also be beneficial when additional views are accessible.

In this chapter, we argue that such problems can be resolved during the propagation process by including geometric cues which help consolidate the geometric reconstruction and the estimation of the local affine transformations, thus avoiding tedious post processing operations in the first place. The use of geometric information in propagation based methods is not new, for instance [42], proposes a splatting inspired approach based on local plane fitting. However, this approach is limited to narrow baselines and requires a relatively large number of images. Our approach, on the other hand, proceeds by fitting small surface patches on which initial 3D points are projected to adjust the matches and the local affine transformations. By operating

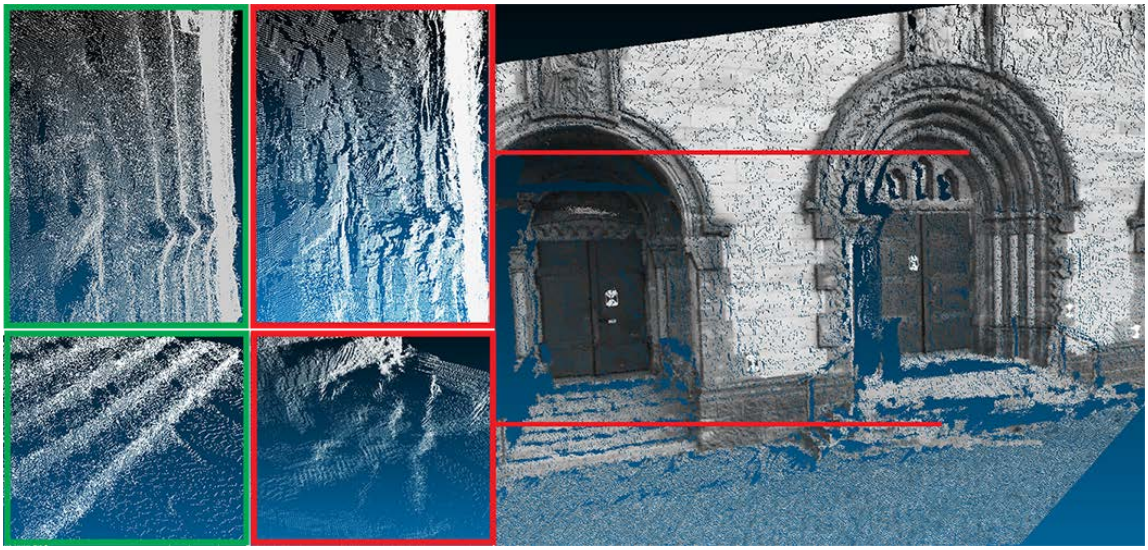


Figure 7-1: Reconstruction from views 3-4 of the Herz-Jesu P8 dataset using [54] (right). Geometric artifacts are clearly visible in the closeups (middle). Whereas, the geometry of the scene is better captured with our approach (left).

in such way, the propagated information is confirmed by both optical and geometrical cues throughout the whole matching process.

## 7.2 Geometry based image match propagation

### 7.2.1 Overview

The central idea of our approach is to couple geometric and optical information in a complementary fashion, in the sense that they correct and confirm each other. In this regard, when a sufficient number of data points are available in a localized region, a surface fitting can be performed and subsequently the matches and their affine transformation will be updated geometrically.

In practice, three main recurrent stages take place in our method after the initial seeds are found using [80]: propagation, region querying, and update by surface

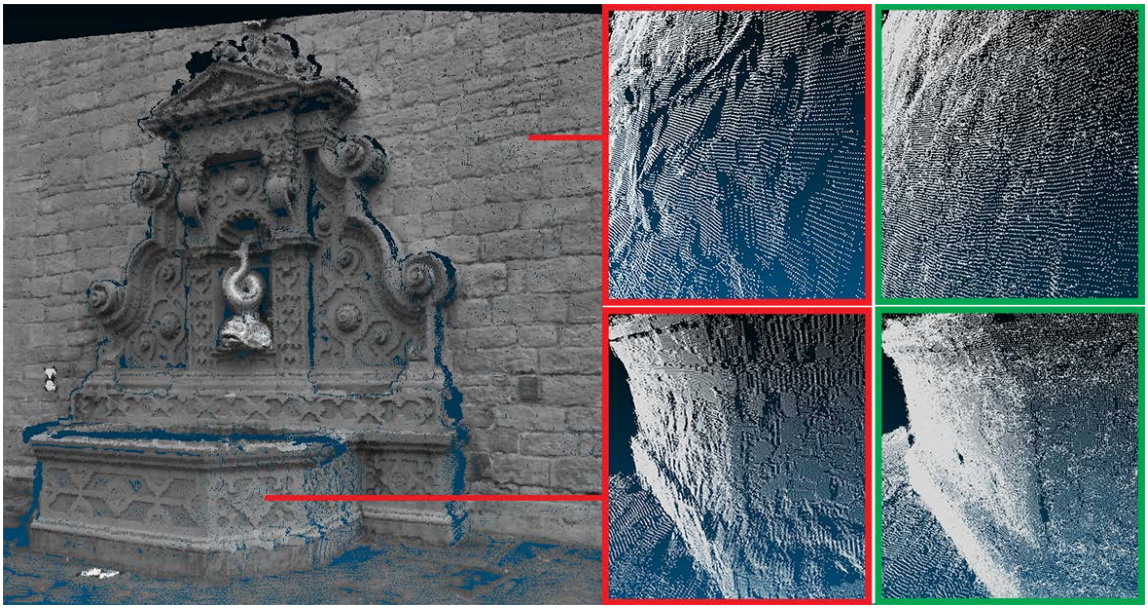


Figure 7-2: Reconstruction from views 2-3 of the Fountain P11 dataset using [54] (left). Geometric artifacts are clearly visible in the closeups (middle). Whereas, the geometry of the scene is better captured with our approach (right).

fitting. In the next paragraphs we briefly describe these stages and we later explain them in greater algorithmic detail in the next sub-sections.

**i) Propagation:** In a propagation step the best seed ($s$) from the queue is selected and new candidates within a neighborhood of $s$ are obtained using its corresponding affine transform.

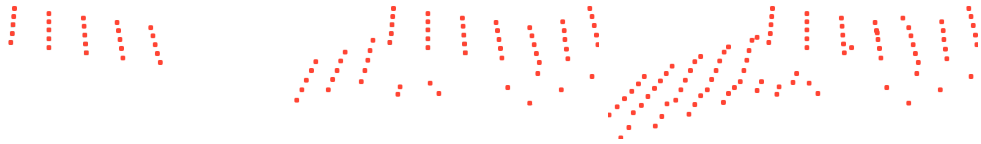
Candidates with correlation scores which pass a minimum threshold and comply with map occupancy and epipolar constraints are added to the queue (see figure 7-3) and are marked as unconfirmed matches.

**ii) Best candidate region querying:** When new matches are found in the propagation step, the search for a suitable region for local surface fitting is initiated. This

search is performed by sliding corresponding windows in the two image planes (depicted by the windows in figure 7-5). Candidate regions where the data points provide enough support for surface fitting are validated and sorted giving preference to the ones with the most confirmed points.

**iii) Local fitting and re-projection:** When a suitable region is found in the previous step a local surface patch fitting is performed. A sub-set of the points representing the core of the region (see figure 7-5), are then projected onto the local surface patch. These new positions are re-projected to the image planes to update the correspondences.

Points that re-project too far in the image planes or yield a large change in their affine transformation remain marked as unconfirmed. Matches which improve their correlation scores are updated and marked as confirmed. This update serves the purpose of correcting point locations as well as steering subsequent propagation (see figure 7-4).

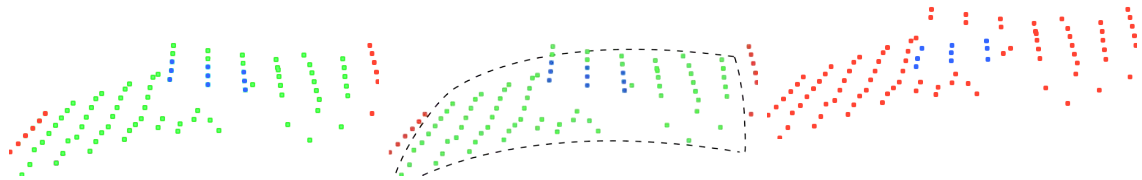


(a) Propagation step n (b) Propagation step n+1 (c) Propagation step n+2

Figure 7-3: Illustration of a few propagation steps before performing any geometric updates.

### 7.2.2 Propagation

In principle, the propagation stage proceeds similarly to the wide baseline matching proposed in [54] but presents three important differences. **First**, newly propagated matches are not defined as final matches; instead, they are stored as unconfirmed



(a) Candidate region selected (b) Local surface fitting (c) New propagation replaces initial points

Figure 7-4: Illustration of the geometric fitting steps. First, the candidate region is selected (a), the support and the core are colored in green and blue respectively. Second, local surface fitting is performed (b). Third, the core points are projected onto the surface (c).

matches. **Second**, the seeds sorting operation gives preference to geometrically confirmed seeds and checks for correlation score in a secondary sorting step. **Third**, new matches stemming from a geometrically confirmed seed can replace previous unconfirmed matches if their correlation score is higher. Note that this second set of conditions do not come into play from the beginning since it takes several propagation steps to obtain a set of geometrically confirmed matches. This categorization of points is depicted in figure 7-5 where blue indicates confirmed points and red indicates unconfirmed (replaceable) ones. In order to fix the ideas, we describe the steps of our propagation stage in algorithms 2 and 3. Note that the 3D positions originating from each pair of matches are obtained at this stage since they are needed for the later region search and surface fitting.

**Input**: Set of seeds with local affine transforms, projection matrices $P$,$P'$
**Output**: Matches, Seeds, 3D point Cloud

```
while Seeds ≠ ∅ do
    Sort(Seeds);
    foreach seed (x, x') do
        Local = ∅ ;
        /* Local stores new candidates                                  */
        foreach (u, u') in N(x, x') do
            if SampsonDistance(u, u') < sd and ((map_1(u) and map_2(u') are
            not filled) or (s is confirmed and correspondence occupying map_1(u)
            and map_2(u') is unconfirmed)) then
                Compute z = ZNCC(u, u') and standard deviations d, d';
                if z > min_correlation and d > t and d' > t then
                    store (u,u',z) in Local;
                end
            end
        end
        sort(Local);
        foreach Pair (u, u') in Local do
            if map_1(u) and map_2(u') are not taken then
                X ← Compute3D(u, u', P, P');
                A_O ← UpdateAffineEstimationOptical(u,u');
                mark map_1(u) and map_2(u') as taken by an unconfirmed point;
                store (u,u',A_O) in Seeds;
                store (u,u',A_O) in Matches;
                store (X) in Cloud;
            end
        end
        while R ←
        BestCandidateRegionQuerying(map_1,map_2,Cloud,Matches) exists do
            FitSurfacePatch(R,Cloud,Seeds,Matches);
        end
    end
end
```

**Algorithm 2:** Main match propagation. Parameter values used in all our experiments: $min_correlation = 0.75$, $t = 2$, $sd = 1$. The FitSurfacePatch is summarized in algorithm 3. The SampsonDistance is defined as in [46] and the Sort function follows the conditions set in 7.2.1

### 7.2.3 Candidate region querying

The goal at this stage is to identify regions with adequate point distribution and density to support a local patch fitting. The region querying is initiated once a set of new unconfirmed points are obtained. In order to steer clear from problematic regions during this search we avoid: i) areas which present large jumps, ii) points which do not remain within their respective patch perimeter when projected on the corresponding images planes. These requirements help prevent fitting and consequentially smoothing sharp features, e.g. the stairs in figure 7-9. On the other hand, we favor regions that present the most confirmed points so that the large bodies of confirmed points expand first instead of creating several isolated clusters of confirmed points. A typical scenario is shown in figure 7-5 where a set of confirmed matches (in blue) is surrounded by a set of unconfirmed points (in red). The green windows represent the support region (outer square) and the core window (inner square). Only unconfirmed points inside the core window will be fit in the next stage. In the same figure, the points inside the orange region represent newly matched points that enable a surface to be fit at the depicted location.

The conditions that define wether surface fitting can be performed are:

- points inside the support and core regions defined on the first image should also lay inside the corresponding support and core regions defined on the second image

- there should be at least one point in the core that was not yet confirmed

- regions inside the support window but to the north, south, east and west of the core window should be sufficiently populated (at least 50%)

- points inside the core window should be dense enough, any $2x2$ window should contain at least one match(pixels)

The sizes of the windows in all of our experiments are $15 \times 15$ for the support and $5 \times 5$ for the core. These windows are defined in the view that presents the most fronto-parallel surface with respect to the image plane and are transformed to the other view using the affine transformation estimation of the point at the center of the window.

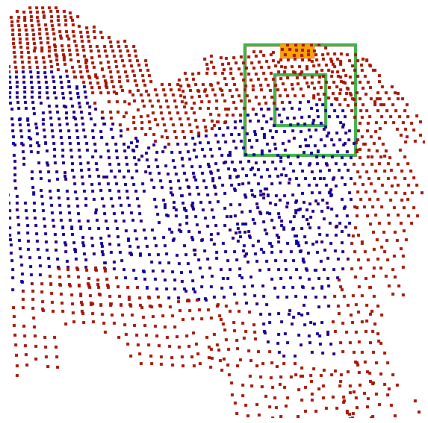


Figure 7-5: Searching for best region to fit a surface patch. Confirmed points are depicted in blue, unconfirmed ones in red. Points in the orange region were obtained in the preceding propagation step. In green, support (outer) and core (inner) windows.

### 7.2.4 Fit surface and update

In this stage, we perform a surface fitting to the points inside the region obtained in the previous stage. For this purpose we use a least squares quadratic polynomial fitting of the form $ax^2 + by^2 + cxy + dx + ey + f$ [7]. Although it is possible to perform higher order fitting, we found in our experiments that quadratic fitting is fairly satisfactory for our purpose. The selected points used in this fitting stage are either part of the core region or of the support region. The core region encompasses a small set of points that are surrounded by a set of support points; this is because points should be fit and projected only when there is enough information to describe the local surface.

During the fitting process, the impact of each point in the least squares formulation can be controlled through a set of weights [7]. We formulate the weights as the product of three values: the correlation score of the corresponding matching points, the inverse 3D distance of the reconstructed point to the center of the region, and the nature of the point (confirmed or unconfirmed). More specifically, $w = z^3(1 + is_confirmed)/distance(X, X_c)$, where $z$ is elevated to the third power in order to stretch the range. $X_c$ is obtained by first finding the central point in $2D$ and then obtaining its corresponding $3D$ location. Once the surface is fit, each $3D$ point $p$ is projected (orthogonally) to a point $p'$ that lies on the fitted surface.

Each newly obtained point $p'$ can then be reprojected to the image planes $I$ and $I'$, obtaining the respective matching image positions $u_r$ and $u'_r$. Furthermore, an estimation of the associated affine transformation is recovered. This affine transformation estimation ($A_G$) is obtained by sampling two additional points around $p'$, projecting them to the image planes and computing the affine transformation that describes their local motion using the re-projection of $p'$ as center of coordinates. Once these new estimations are obtained, we first require that the new 2D matching points induce only a small adjustment of the initial matching positions. Namely, $distance(u_r, u) < \gamma$ and $distance(u, u'_r) < \gamma$ (with $\gamma = 1.5$ pixels used in all our experiments). In this manner we ensure that points near surface edges do not drift away from their original location. Next, we verify if $A_G$ does not represent a large change in transformation estimation with respect to the current estimation $A$. At the same time we need to verify that $A_G$ does not yield a large shear. Specifically, we verify if $\theta \leq det(A)/det(A_G) \leq 1/\theta$, with $\theta = 0.5$ for all of our cases. We then define the eigen-values rations $eigratio_g = \frac{eig_1(A_G)}{eig_2(A_G)}$ and $eigratio = \frac{eig_1(A)}{eig_2(A)}$ and verify if $\theta \leq eigratio_g \leq 1/\theta$ and if $\theta \leq eigratio_g/eigratio1/ \leq \theta$. A point will remain as unconfirmed if it does not fulfill all of these requirements. Note that the common use of $\theta$ for all the previous tests is not required. Independent values can be used for the three verifications but in all of our experiments the same value was used.

The steps described above are summarized in algorithm 3.

```
Input: region R, list of Matches, point Cloud, list of Seeds
Output: Updated: Cloud, Matches and Seeds
w ← (z^3 × 1/distance(X, X_c) × (1 + is_confirmed));
perform least squares surface fitting to obtain surface Q;
foreach 3D point p inside the core of region R do
    obtain p' by projecting p to surface Q;
    if distance(p', p) < ε then
        p'_1 ← re-project p' to I;
        p'_2 ← re-project p' to I';
        A_G ← compute affine transformation using Q and p';
        if A_G does not represent a large change from A and does not represent
        too much distortion then
            Obtain normalized windows using A_G;
            z_new ← ZNCC(p'_1, p'_2);
            if z_new > z_old then
                UpdateCloud(p');
                UpdateMatches(p'_1, p'_2, A_G);
                UpdateSeeds(p'_1, p'_2, A_G);
                mark map_1(p'_1) and map_2(p'_2) as taken by a confirmed point;
            end
        end
    end
end
```

**Algorithm 3:** *FitSurfacePatch* algorithm that performs matching correction by surface fitting and re-projection

## 7.3 Results

We tested our algorithm on various data sets comprising standard benchmarks as well as in-house acquired data. Typical results of our approach compared to those of Kannala and Brandt [54], Brox and Malik [19] and Tola et al. [124] are shown for the views 2-3 of the Fountain-P11 dataset (figures 7-6 and 7-7), and for the views 3-4 of the Herz-Jesu-P8 dataset (figures 7-8 and 7-9). Both datasets are provided in [120]. We benchmarked our results measuring depth estimation error with respect to the ground truth, in a similar way as in [142]. The benchmarked results are presented in figures 7-6 and 7-8 as error vs occupancy histogram graphs and as color coded depth estimation errors. We use the code provided by the authors of [54] to perform comparisons and we used similar values for the common parameters of the algorithms. We also use the code provided by the author of the variational matching method [19]. For the case of [124], we use the code provided by the author to compute the DAISY dense descriptor in combination with our own implementation of a descriptor matching method that searches for matches within rectified images. Since such approach does not accurately represents what is proposed in [124], we also compare our results to those presented in the later paper, following their own evaluation metrics (see figure 7-12). Notice that in both quantitative evaluations our method achieves better results. The running time for our propagation approach is 17 minutes for the case of two images of the Fountain scene and the Herz-Jesu scene, which contain 6MPx.

In all cases our approach performs better than the state of the art pairwise propagation approach [54] and the other methods here tested. The depth error benchmarks presented in figures 7-6 and 7-8 clearly show that our method presents more accurate and more complete results.

In figure (7-6-top-right) it can be observed that our method attains 74.2% of the viewed Fountain scene with an error of less than 0.1; while the method of Kannala and Brandt achieves only 68.3%. In the same error vs occupancy graph we oppose

our results to the combination of [54] with an MLS [7] in post-processing. This combination leads only to a small improvement for errors in the lower range but an overall decay in the quality of the results is observed.

In figure (7-8-top-right) our method's score for errors lower than 0.1 in the viewed Herz-Jesu scene is 63.6%. The method of [54] as well as its combination with MLS post-processing yield only 58.4% and 50.9% of occupancy respectively.

The closeups to the fountain result presented in figure 7-7 illustrate the improvements gained by using our method. Our results do not exhibit the choppy characteristic of previous work and present a clearer and more accurate structure of the scene. The closeups to the Herz-Jesu result presented in figure 7-9 not only show that the quality is improved but also the total coverage. In this example the main stairs in the scene are accurately reconstructed thanks to the continuous interpretation and improvement of the matching results that lead to better propagations.

We also performed a comparison using two views of our own face data-set that comprises 1.3 mega-pixels images and two views (7 and 15) of the warrior data-set from [33]. Results are presented in figures 7-10 and 7-11 were uncolored closeups are presented next to the full views of the results. In both cases the results are significantly better when our approach is used.

Finally, in figure 7-13 we show a quantitative comparison of our distortion driven variational method 5 against the propagation based matching technique proposed in this chapter.

## 7.4 Conclusion and discussion

We have presented a propagation approach which congruently uses optical and geometric information to steer the propagation process. Although, many of underlying ideas are fairly simple, they lead to significant improvements both in the accuracy and completeness. Furthermore, as it could be argued that performing an equivalent

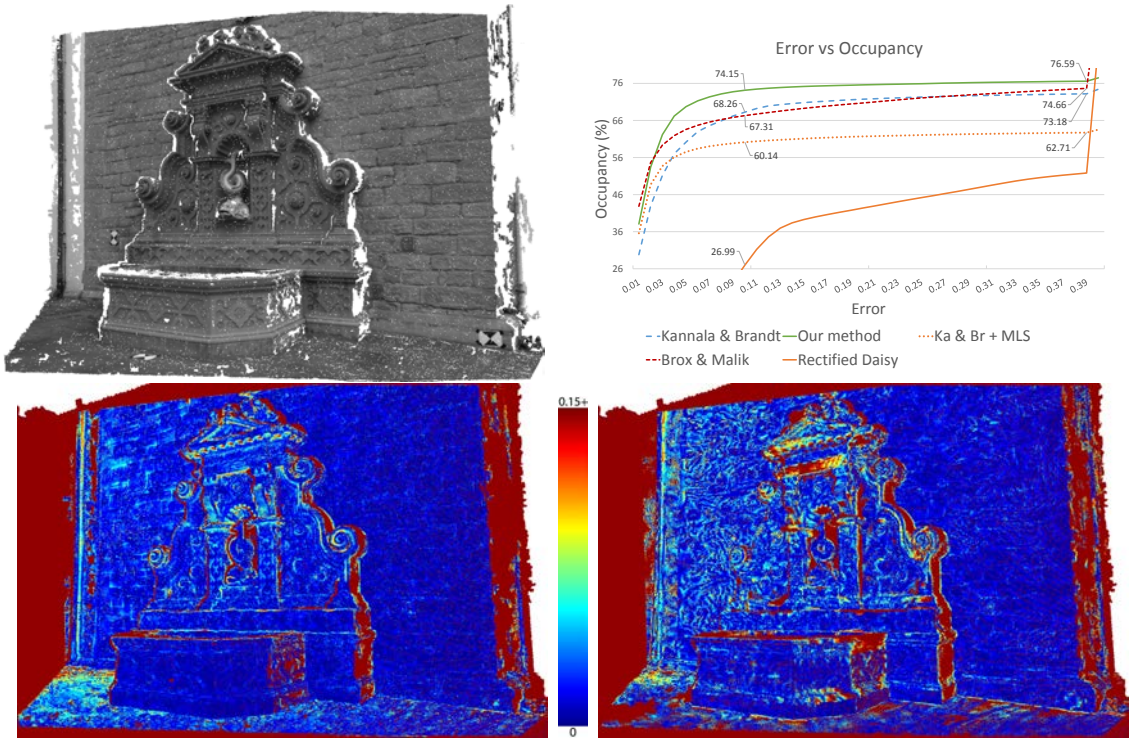



Figure 7-6: Results of our approach on the Fountain P11 views 2-3 (top-left). On the top-right, a chart illustrating the depth error vs occupancy of our algorithm, [54], [19], [124] (rectified) and [54] with MLS as post-processing.On the bottom color coded depth error using our algorithm (left) and using [54] (right).

MLS fitting as post processing would yield equivalent results, our comparisons to such a scenario, suggest that actually the results get worse. The reason being that in our case the subsequent matches depend on the fitting, whereas the use of fitting in post processing cannot alter how the matches are propagated. Finally, we have compared the use of variational matching against the use of the propagation based reconstruction proposed in this chapter. In both scenes the propagation-based methods outperform variational techniques, with our method providing the results in the upper curve.

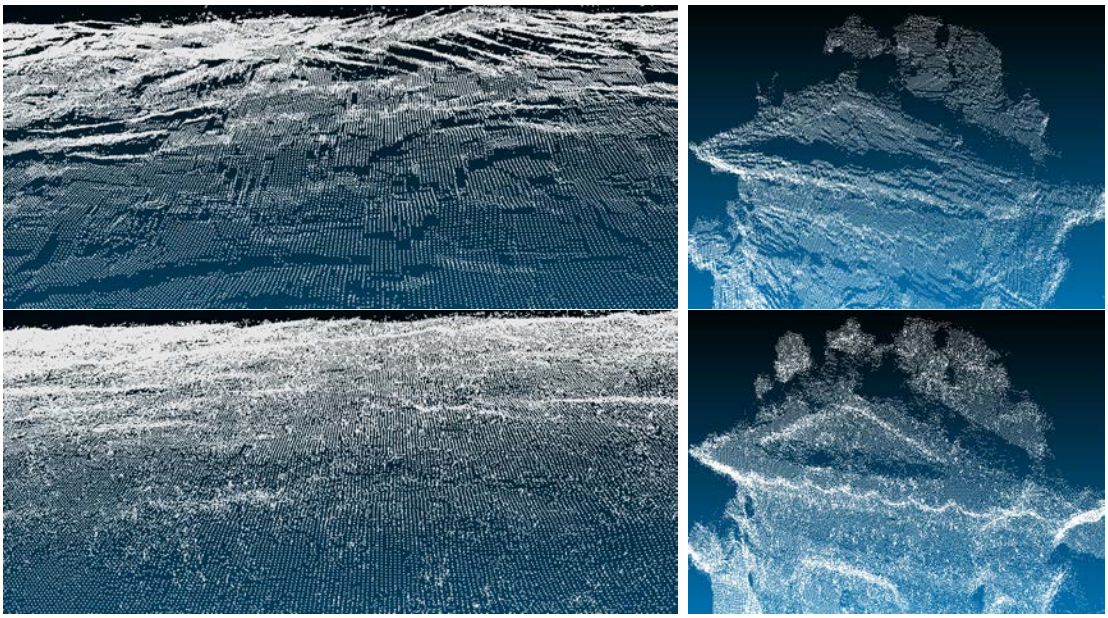


Figure 7-7: Closeups on portions of the wall and fountain-top for the Fountain P11 views 2-3 using [54] (top) and using our method (bottom)

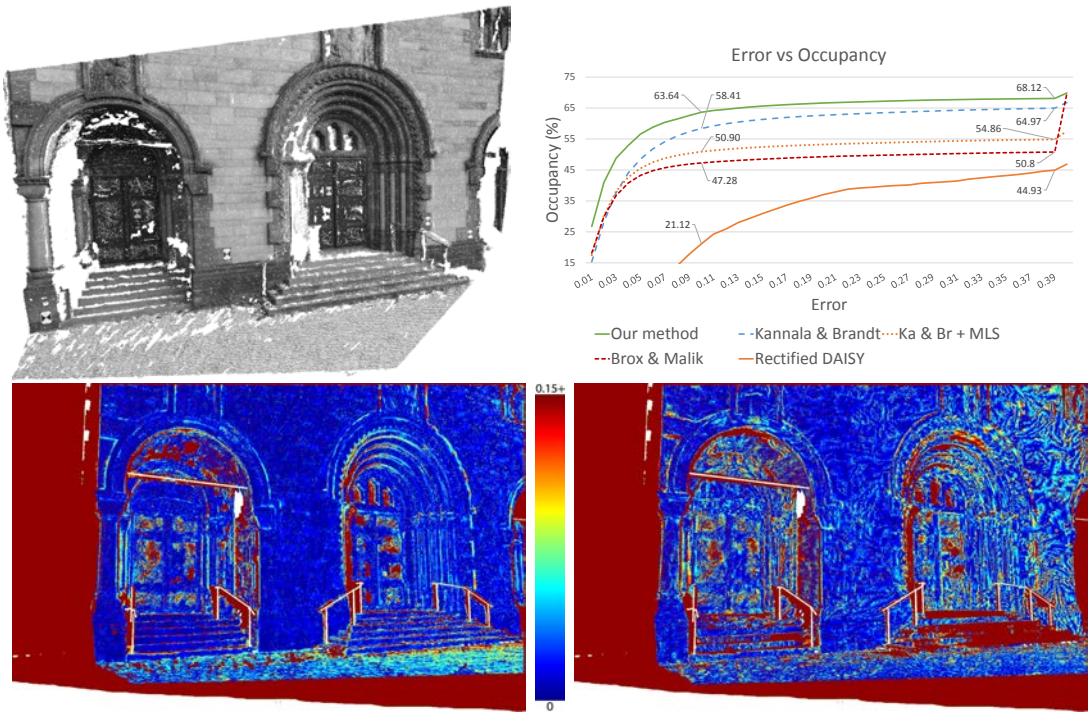



Figure 7-8: Result of our approach on the Herz-Jesu P8 views 3-4 (top-left). On the top-right, a chart illustrating the depth error vs occupancy of our algorithm, [54], [19], [124] (rectified) and [54] with MLS as post-processing. On the bottom, color coded depth error using our algorithm (left) and using [54] (right).

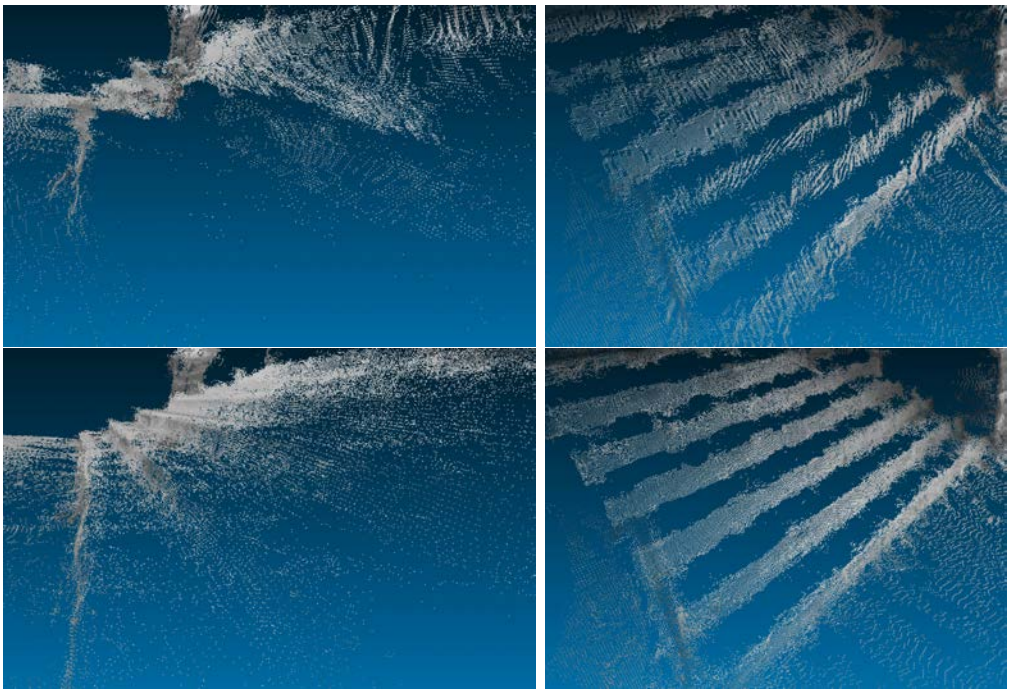


Figure 7-9: Closeups on to results on Herz-Jesu P8 views 3-4 using [54] (top) and using our method (bottom). A closeup to a portion of the main stairs (left) and to the left-most stairs in the scene (right)

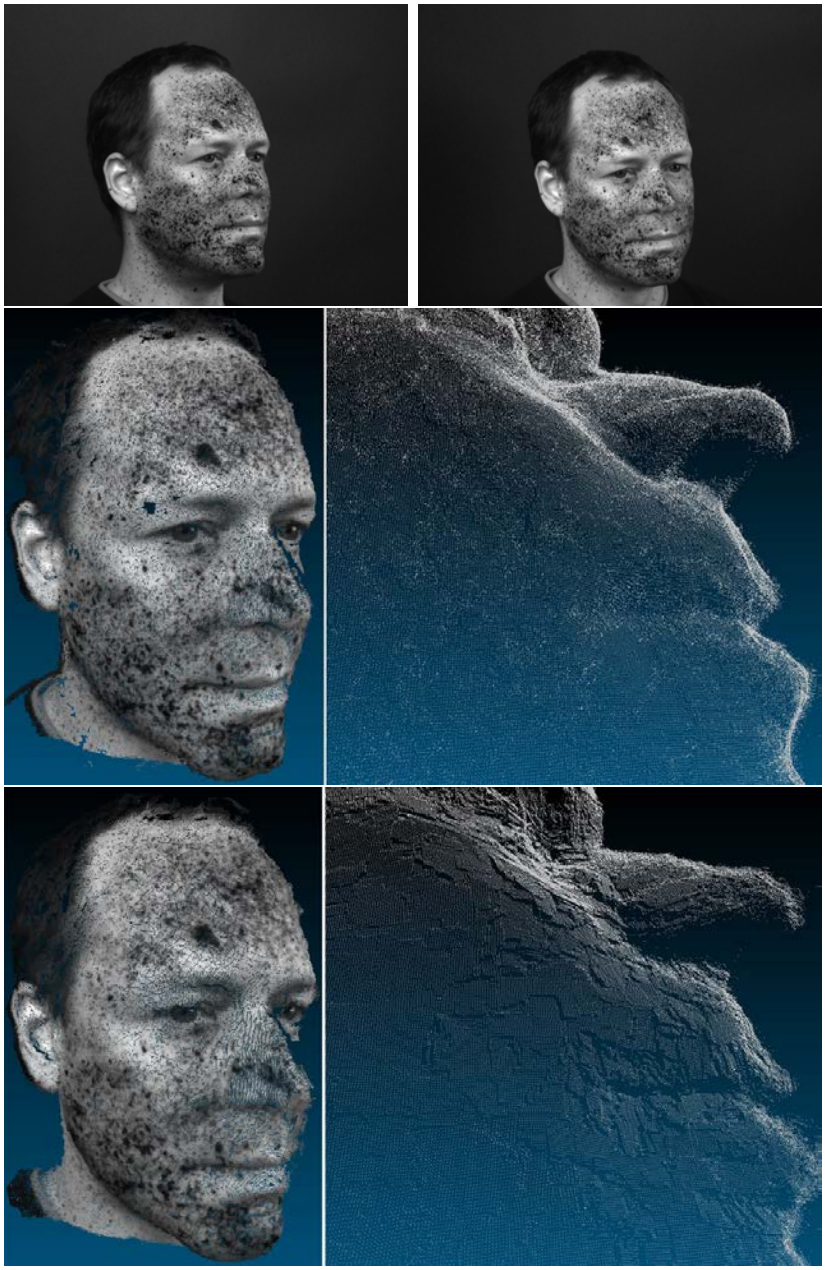


Figure 7-10: Results on two views of our face dataset. The two views used (top), our result with a closeup to the cheek (middle), results using [54] (bottom)

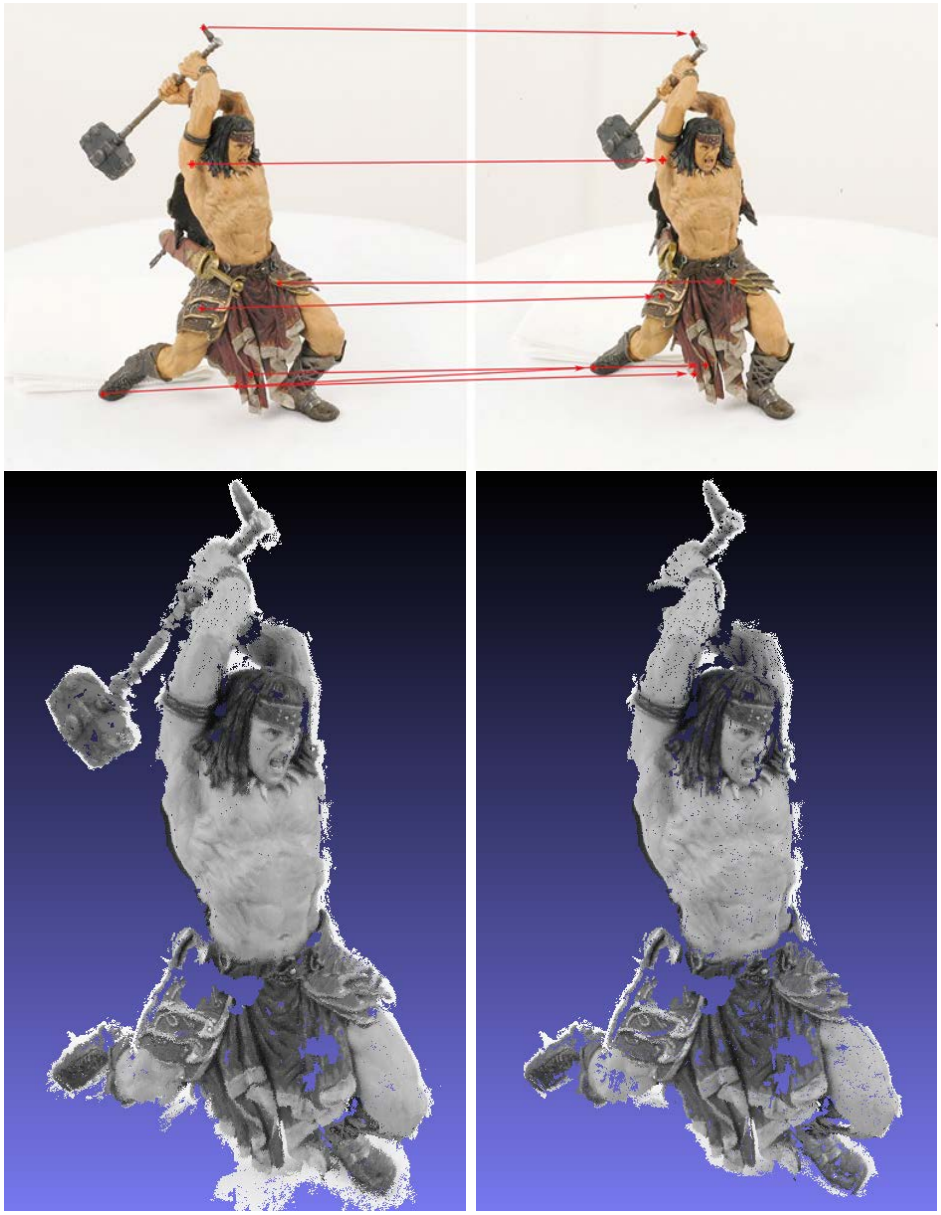


Figure 7-11: Results on views 7 and 15 of the warrior dataset from [33]. The two views used are showed at the top row along with the 7 features matched. On the bottom row, our result (left) and results using [54] (right). Notice that our method returns less holes and it is able to reconstruct the warrior's hammer.

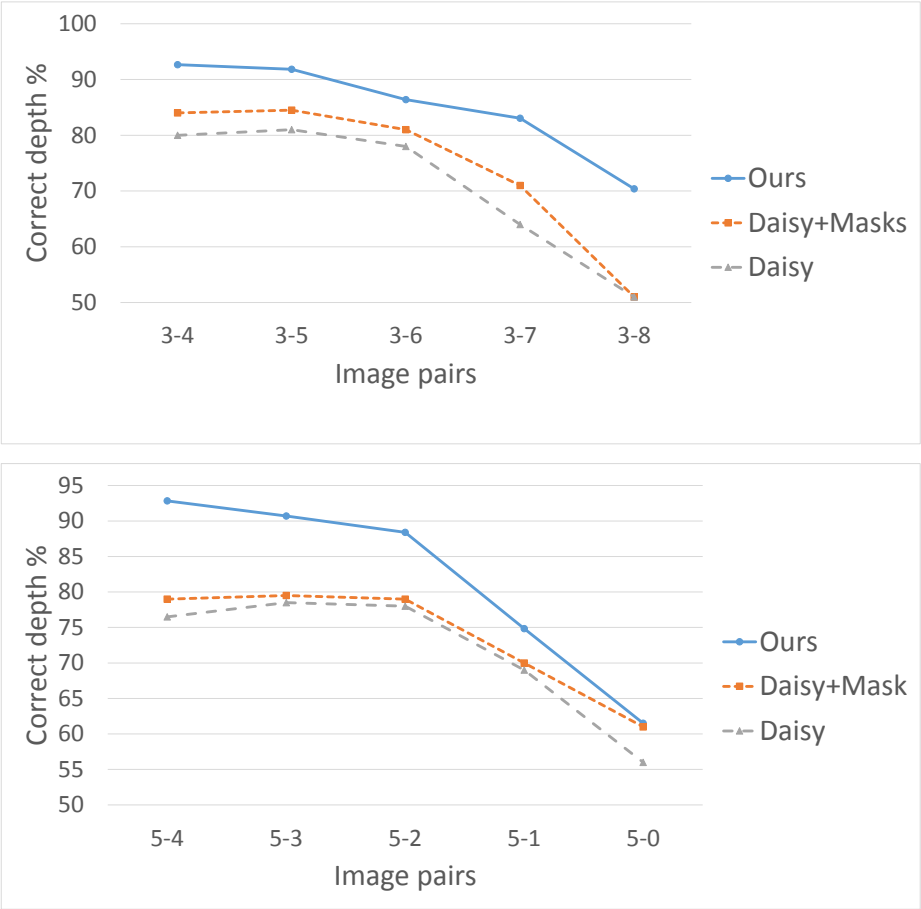
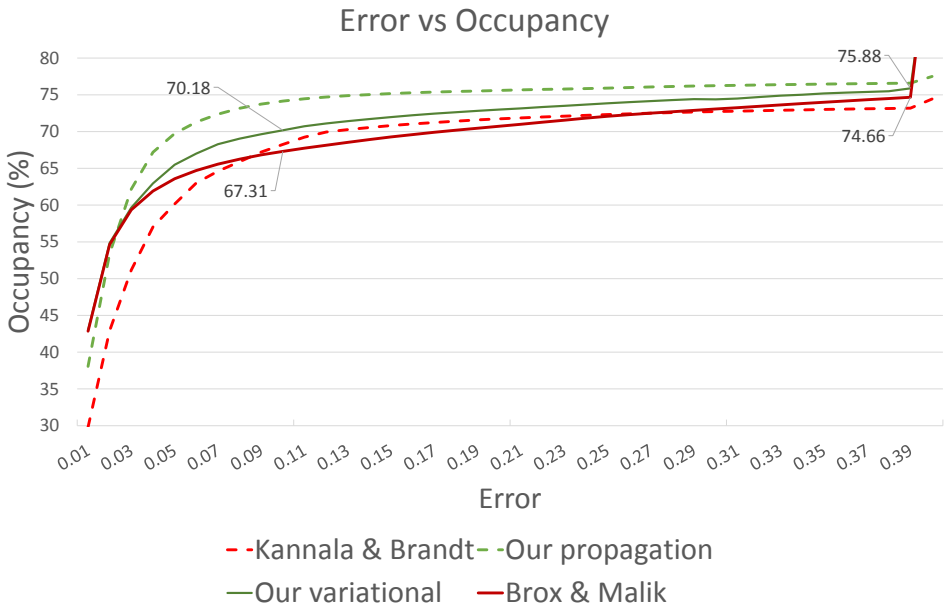



Figure 7-12: Comparisons of our results to those of DAISY [124]. Results on the Fountain scene (top) and the Herz-Jesu scene (bottom) for pairs of images with increasing baselines. The charts present the percentage of correct depth estimations for each pair of views. A depth estimation is considered correct if it presents and error of less than 1% of the scene's depth range [124] when compared to the laser scanned data.



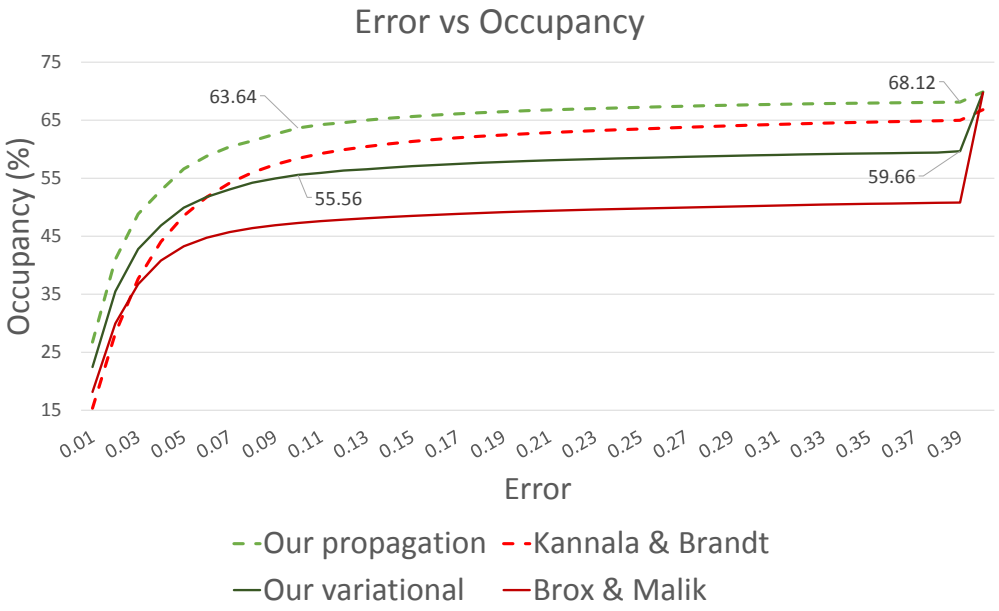




Figure 7-13: Comparisons between propagation based methods and variational methods. The methods used are: our variational (explained in this chapter), the method of Kannala and Brandt [54], our variational method proposed in chapter 5 and the method of Brox and Malik [19]

# Chapter 8

# Conclusion and perspective

## 8.1 Conclusion

Throughout this work we have shown that information about local geometry can be used to obtain more accurate 3D matching.

We have first explored variational matching methods and observed that well-behaved results can be obtained due to error equi-distribution. Nevertheless, we have shown, through quantitative evaluations, that results can be improved further if the formulation includes information about scene geometry (e.g. epipolar constraints). This improvement, however, is limited and not locally adaptive to important changes in scene depth. We have closely analyzed the effects of abrupt changes in scene depth and determined that classical variational matching techniques not only fail to accurately recover these changes but they also destroy surrounding areas through an error bleeding effect of the regularizer. In our analysis, we observed that these areas present a large distortion in the resulting $3D$ grid that we obtain by projecting the $2D$ grid given in the image plane. The local $3D$ grid distortions can be accurately characterized and introduced to modify the regularizer in an anisotropic manner, effectively pushing grid lines away from problematic regions and reducing the bleeding

effect. We have shown, through experiments, that this proposed technique yields improved results with respect to classical variational matching techniques.

Continuing in the context of variational methods, we proposed to obtain multi-view reconstructions by merging pairwise variational results. We merge these results using information about the $2D$ grid deformation to select the best contributions across neighboring views allowing to reduce data redundancy before dealing with all the $3D$ data. Unlike most of related work, which seek to merge pairwise results considering all the $3D$ data at hand, we proposed to resolve a large number of outliers and redundancies prior to estimating the final spatial point cloud.

All of our contributions, in the context of variational matching for 3D reconstruction, are systematic since they are achieved by means of simple, yet principled, characterizations of geometric deformations which can be easily reused in other related methods.

While variational methods provide well behaved results with equi-distributed errors, we have observed that local methods based on match propagation provide results that adapt closely to varying $3D$ structures since they operate locally. We have seen that the matching approach presented in [54], allows accurate 3D reconstructions. The problem with this method, and with local methods in general, is that they can present several outliers and a noisy nature. Methods such as [34, 35] rely on post processing to reduce the mentioned shortcomings but they require a large number of images and we have shown that they provide less complete results. We have then argued that there is much more that can be used when matching a pair of images and even before considering a multi-view strategy. To this end, we formulated a propagation approach which congruently performs matching while assuming that the surface at hand is locally smooth. We have used this assumption by performing local surface fittings and projections to correct point locations. Although, many of the underlying ideas of this propagation method are fairly simple, we have shown that they lead to significant and quantifiable improvements in terms of accuracy and completeness of

the results.

In section 7.3 we have compared variational and propagation-based matching techniques and shown that our propagation-based reconstruction provides the best results. Nevertheless, there could be scene setups where variational matching could be recommended. For example, in the case of high scene overlap in the images or in the presence of very smooth objects. In the following table we list some of pros and cons of using one of the two methods proposed in this thesis.

| Method | Pros | Cons |
|---|---|---|
| Variational matching | -Formulated globally<br>-Easy to parallelize (e.g. [144]) | -Parameter tuning required<br>-Matches searched everywhere<br>-Hard to scale |
| Propagation | -Recovers discontinuities automatically<br>-Computes results only where meaningful | -Edges not well defined<br><br>-Hard to parallelize |

Table 8.1: Pros and cons of variational and propagation based $3D$ reconstruction

## 8.2 Perspective

In section 7.3 we have shown how propagation based matching can outperform variational matching techniques in the case of wide baseline setups. However, in the same section, we have pointed out that the current solution is slow compared to other propagation based techniques (approximately three times that of our C++ implementation of [54]) and compared to variational matching techniques (approximately five times that of our variational technique in chapter 5). This is related to the several checks and operations at each propagation step. There is also the fact that propagation based matching uses a priority queue that is difficult to parallelize and, therefore, all operations work in a sequential manner. Furthermore, the current feature matching technique [81] used in our propagation based matching provides only a few initial matches (seeds), making it difficult to propagate from multiple fronts in

the scene. Therefore, we are currently working on a solution that uses the feature matching technique proposed in [20]; which provides many more matching points (see sections 3.3.3 and 3.5). Using this feature matching technique, we would like to select multiple, non-adjacent seeds and propagate them in parallel. Additionally, we believe that the accuracy of the results can be further improved by allowing seeds to only propagate to relevant regions. For example, in figure 8-1 (first shown in section 7.3) we observe that our results for the floor close to the bottom of the stairs present a larger error when compared to the results from [54]. This occurs because the stairs are properly matched in our results and propagation can then occur from these stairs to the floor; effectively starting with less accurate approximations for the affine transformation. Therefore, we would like to favor propagations that start with closer initial estimations.

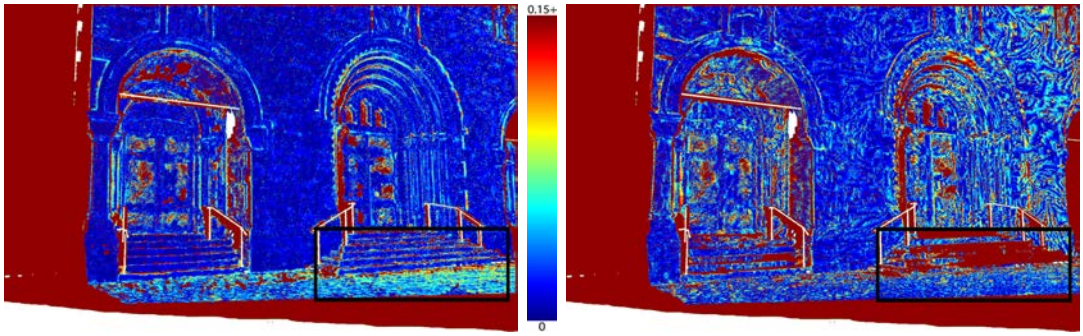



Figure 8-1: Color coded depth error using our algorithm (left) and using [54] (right).

In figure 8-2, we show an example of using the LDAHASH feature matches. In this figure, the complete propagation that starts from a feature is illustrated with one color; so each colored region represents an independent propagation. In the left figure, we observe a lower number of regions. We can also see that propagations can jump from a wall to a floor (e.g. from the base of the fountain to the floor). While this could be useful in cases where there are only a few seeds, we would like to limit this behaviour to cases where it is absolutely needed. In the same figure, in the

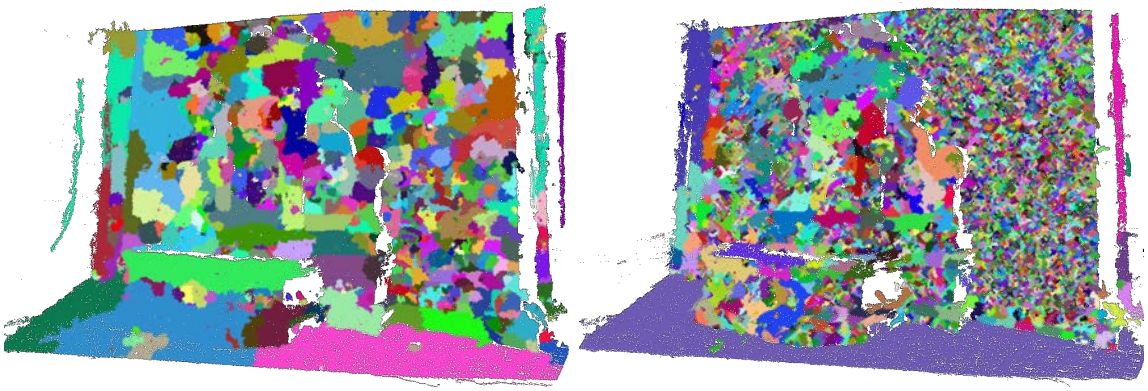


Figure 8-2: Color coded propagation for the fountain scene where each color represent matches propagated from a seed. To the left, propagation results using the method proposed in Chapter 7. To the right, sample propagation of our envisioned method outlined in this section.

image to the right we observe that there are many more seeds and that propagations from different surfaces are limited (e.g. matches on the base of the fountain do not propagate to the floor). Finally, preliminary results show that this match propagation works better in a multi-view setup. Not by merging pairwise results but by matching multiple views at the same time. In this way, mismatches and outliers due to repetitive patterns, low textured regions and image noise can be reduced. Furthermore, the prioritized propagation can take into account that matches which include multiple views convey information with higher confidence.

# Publications

1. Patricio Galindo and Rhaleb Zayer. *Complementary geometric and optical information for match-propagation-based 3D reconstruction.* Asian Conference on Computer Vision, Singapore, 2014.

2. Patricio Galindo and Rhaleb Zayer. *Distortion driven variational multi-view reconstruction.* International Conference on 3D Vision, Tokyo, Japan, 2014.

3. Kun Liu, Patricio Galindo and Rhaleb Zayer. *Sphere Packing Aided Surface Reconstruction for Multi-View Data.* International Symposium on Visual Computing, Las Vegas-USA, 2014.

# Bibliography


[1] A.E. Abdel-Hakim and A.A. Farag. Csift: A sift descriptor with color invariant characteristics. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1978–1983. IEEE, 2006.

[2] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez. Symmetrical dense optical flow estimation with occlusions detection. In *Computer Vision—ECCV 2002*, pages 721–735. Springer, 2002.

[3] L. Alvarez, R. Deriche, J. Sánchez, and J. Weickert. Dense disparity map estimation respecting image discontinuities: A {PDE} and scale-space based approach. *Journal of Visual Communication and Image Representation*, 13(1 - 2):3 – 21, 2002.

[4] F. Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183, 1954.

[5] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[6] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[7] R.E. Barnhill, G. Farin, M. Jordan, and B.R. Piper. Surface/surface intersection. *Computer Aided Geometric Design*, 4(1):3–16, 1987.

[8] J. Barron and R. Klette. Quantitative color optical flow. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 4, pages 251–255 vol.4, 2002.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[10] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

[11] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Computer Vision—ECCV'92*, pages 237–252. Springer, 1992.

[12] M. J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Comput. Vis. Image Underst.*, 63(1):75–104, January 1996.

[13] J.-Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5, 2001.

[14] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.

[15] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, 31(138):333–390, 1977.

[16] T. Brox. *From pixels to regions: partial differential equations in image analysis.* PhD thesis, 2005.

[17] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 41–48, 2009.

[18] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In Tomás Pajdla and Jiří Matas, editors, *Computer Vision - ECCV 2004*, volume 3024 of *Lecture Notes in Computer Science*, pages 25–36. Springer Berlin Heidelberg, 2004.

[19] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):500–513, 2011.

[20] M. M. Bronstein C. Strecha, A. M. Bronstein and P. Fua. LDAHash: Improved Matching with Smaller Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1), 2012.

[21] J. Cech and R. Sara. Efficient sampling of disparity space for fast and accurate matching. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, june 2007.

[22] Q. Chen and G. Medioni. A volumetric stereo matching method: Application to image-based modeling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1029–1034, Fort Collins, United States, June 1999.

[23] O. Chum, T. Werner, and T. Pajdla. Joint orientation of epipoles. In *BMVC*, pages 1–10, 2003.

[24] S.D. Conte and D.E. Boor. *Elementary Numerical Analysis: An Algorithmic Approach*. McGraw-Hill, New York, 1972.

[25] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[26] J.G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision research*, 20(10):847–856, 1980.

[27] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *International journal of computer vision*, 10(2):101–124, 1993.

[28] P. Eisert. Reconstruction of volumetric 3d models. *3D Videocommunication: Algorithms, Concepts and Real-Time Systems in Human Centred Communication*, pages 133–150, 2005.

[29] V. Ferrari, T. Tuytelaars, and L. V. Gool. Wide-baseline multiple-view correspondences. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–718. IEEE, 2003.

[30] V. Ferrari, T. Tuytelaars, and L. J. Van Gool. Simultaneous object recognition and segmentation by image exploration. In Tomás Pajdla and Jiri Matas, editors, *Computer Vision - ECCV 2004, 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I*, volume 3021 of *Lecture Notes in Computer Science*, pages 40–54. Springer, 2004.

[31] M.A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[32] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

[33] Y. Furukawa and J. Ponce. 3d photography dataset. `http://www.cse.wustl.edu/~furukawa/research/mview/index.html`. Accessed: 2014-05-05.

[34] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, june 2007.

[35] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362–1376, 2010.

[36] D. Gallup, J-M Frahm, P. Mordohai, and M. Pollefeys. Variable baseline/resolution stereo. In *CVPR*, pages 1–8. IEEE, 2008.

[37] P. Gargallo and P. Sturm. Bayesian 3d modeling from images using multiple depth maps. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 885–891. IEEE, 2005.

[38] J-M Geusebroek, R. Van den Boomgaard, A.W.M. Smeulders, and H. Geerts. Color invariance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(12):1338–1350, 2001.

[39] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *ICCV*, pages 1–8, 2007.

[40] A. Gruen. Adaptive least squares correlation: a powerful image matching technique. *South African Journal of Photogrammetry, Remote Sensing and Cartography*, 14(3):175–187, 1985.

[41] A. Guiducci. Corner characterization by differential geometry techniques. *Pattern Recognition Letters*, 8(5):311–318, 1988.

[42] M. Habbecke and L. Kobbelt. A surface-growing approach to multi-view stereo reconstruction. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[43] R.W. Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950.

[44] R.M. Haralick and L.G. Shapiro. Image segmentation techniques. In *1985 Technical Symposium East*, pages 2–9. International Society for Optics and Photonics, 1985.

[45] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[46] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[47] Horst W. Haussecker and David J. Fleet. Computing optical flow with physical models of brightness variation. In *CVPR*, pages 2760–2767, 2000.

[48] Vu Hoang Hiep, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *CVPR*, pages 1430–1437, 2009.

[49] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):328–341, 2008.

[50] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[51] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.

[52] I Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[53] T. Kanade, O. Amidi, and Q. Ke. Real-time and 3d vision for autonomous small and micro air vehicles. In *Decision and control, 2004. CDC. 43rd IEEE conference on*, volume 2, pages 1655–1662. IEEE, 2004.

[54] J. Kannala and S.S. Brandt. Quasi-dense wide baseline matching using match propagation. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[55] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[56] J. Kelly. Operation crossbow: How 3d glasses helped defeat hitler, 2011.

[57] P. M. Knupp. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Analysis and Design*, 39(3):217 – 241, 2003.

[58] J.J. Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984.

[59] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Computer Vision—ECCV 2002*, pages 82–96. Springer, 2002.

[60] C. Kondermann, D. Kondermann, B. Jähne, and C. Garbe. *An adaptive confidence measure for optical flows based on linear subspace projections*. Springer, 2007.

[61] P. Koskenkorva, J. Kannala, and S.S. Brandt. Quasi-dense wide baseline matching for three views. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, pages 806–809, Washington, DC, USA, 2010. IEEE Computer Society.

[62] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[63] P. Lamon, C. Stachniss, R. Triebel, P. Pfaff, C. Plagemann, G. Grisetti, S. Kolski, W. Burgard, and R. Siegwart. Mapping with an autonomous car. In *IEEE/RSJ IROS Workshop: Safe Navigation in Open and Dynamic Environments*, volume 26. Citeseer, 2006.

[64] M. Lhuillier and L. Quan. Match propagation for image-based modeling and rendering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(8):1140–1146, 2002.

[65] M. Lhuillier and L. Quan. Quasi-dense reconstruction from image sequence. In *Proceedings of the 7th European Conference on Computer Vision-Part II*, ECCV '02, pages 125–139, London, UK, UK, 2002. Springer-Verlag.

[66] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):418–433, March 2005.

[67] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, 1993.

[68] T. Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

[69] T. Lindeberg. Direct estimation of affine image deformations using visual front-end operations with automatic scale selection. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 134–141. IEEE, 1995.

[70] T. Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.

[71] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-d shape cues from affine deformations of local 2-d brightness structure. *Image and vision computing*, 15(6):415–434, 1997.

[72] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, ECCV '08, pages 28–42, Berlin, Heidelberg, 2008. Springer-Verlag.

[73] J. Liu and R. Hubbold. Automatic camera calibration and scene reconstruction with scale-invariant features. In *Advances in Visual Computing*, volume 4291 of *Lecture Notes in Computer Science*, pages 558–568. Springer Berlin Heidelberg, 2006.

[74] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[75] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[76] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Patrick J. Hayes, editor, *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI 81), Vancouver, BC, Canada, August 1981*, pages 674–679. William Kaufmann, 1981.

[77] O. Mac Aodha, A. Humayun, M. Pollefeys, and G.J. Brostow. Learning a confidence measure for optical flow. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(5):1107–1120, 2013.

[78] S. Marčelja. Mathematical description of the responses of simple cortical cells*. *JOSA*, 70(11):1297–1300, 1980.

[79] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[80] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Computer Vision—ECCV 2002*, pages 128–142. Springer, 2002.

[81] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[82] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[83] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, November 2005.

[84] Y. Mileva, A. Bruhn, and J. Weickert. Illumination-robust variational optical flow with photometric invariants. In *Pattern Recognition*, pages 152–162. Springer, 2007.

[85] O. Monga. An optimal region growing algorithm for image segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 1(03n04):351–375, 1987.

[86] H.P. Moravec. Towards automatic visual bbstacle avoidance. In *International Conference on Artificial Intelligence (5th: 1977: Massachusetts Institute of Technology)*, 1977.

[87] J.-M. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[88] D. D Morris and T. Kanade. Image-consistent surface triangulation. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 332–338. IEEE, 2000.

[89] K.P. Murphy, Y. Weiss, and M.I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc., 1999.

[90] O. Naoya. Optical flow detection by color images. In *Image Processing (ICIP), IEEE International Conference on*. IEEE, 1989.

[91] P.J. Narayanan, P.W. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *Computer Vision, 1998. Sixth International Conference on*, pages 3–10. IEEE, 1998.

[92] K. Neckeis. Fast local estimation of optical flow using variational and wavelet methods. In Władysław Skarbek, editor, *Computer Analysis of Images and Patterns*, volume 2124 of *Lecture Notes in Computer Science*, pages 349–356. Springer Berlin Heidelberg, 2001.

[93] S. Negahdaripour. Revised definition of optical flow: integration of radiometric and geometric cues for dynamic scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(9):961–979, 1998.

[94] Y. Ohtake, A. Belyaev, and H.-P. Seidel. An integrating approach to meshing scattered point data. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 61–69. ACM, 2005.

[95] A. Ostrowski. *On the linear iteration procedures for symmetric matrices*. 1954.

[96] M. Otte and H. Nagel. Optical flow estimation: Advances and comparisons. In Jan-Olof Eklundh, editor, *ECCV*, volume 800 of *Lecture Notes in Computer Science*, pages 49–60. Springer Berlin / Heidelberg, 1994.

[97] G.P. Otto and T.K.W. Chau. A 'region-growing' algorithm for matching of terrain images. *Image and Vision Computing*, 7(2):83 – 94, 1989.

[98] D.W. Peaceman and H.H. Rachford. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial & Applied Mathematics*, 3(1):28–41, 1955.

[99] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, 1990.

[100] M. Proesmans, L. Gool, E. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In J.-O. Eklundh, editor, *ECCV*, volume 801 of *Lecture Notes in Computer Science*, pages 294–304. Springer Berlin Heidelberg, 1994.

[101] A.P. Rockwood and J. Winget. Three-dimensional object reconstruction from two-dimensional images. *Computer-Aided Design*, 29(4):279–285, 1997.

[102] K. Rohr. Recognizing corners by fitting parametric models. *International journal of computer vision*, 9(3):213–230, 1992.

[103] S. Roth, V. Lempitsky, and C. Rother. Discrete-continuous optimization for optical flow estimation. In D. Cremers, B. Rosenhahn, A.L. Yuille, and F.R. Schmidt, editors, *Statistical and Geometrical Approaches to Visual Motion Analysis*, volume 5604 of *Lecture Notes in Computer Science*, pages 1–22. Springer Berlin Heidelberg, 2009.

[104] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[105] Zhaohui S. A three-frame approach to constraint-consistent motion estimation. In *ICPR*, volume 1, pages 35–38, 2006.

[106] N. Sabater, S. Leprince, and J. Avouac. Contrast invariant and affine sub-pixel optical flow. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 53–56. IEEE, 2012.

[107] J. Sanchez, N. Monzon, and A. Salgado. Robust Optical Flow Estimation. *Image Processing On Line*, 3:252–270, 2013.

[108] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *Computer Vision—ECCV 2002*, pages 414–431. Springer, 2002.

[109] D.C. Schneider, M. Kettern, A. Hilsmann, and P. Eisert. A global optimization approach to high-detail reconstruction of the head. In *VMV'11*, pages 9–15, 2011.

[110] R. Sedgewick. *Algorithms (2Nd Ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1988.

[111] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006.

[112] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593 –600, jun 1994.

[113] D. Sibbing, M. Habbecke, and L. Kobbelt. Markerless reconstruction and synthesis of dynamic facial expressions. *Computer Vision and Image Understanding*, 115(5):668 – 680, 2011. Special issue on 3D Imaging and Modelling.

[114] S.M. Smith and J. M. Brady. Susan—a new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.

[115] G. Socher, T. Merz, and S. Posch. 3-d reconstruction and camera calibration from images with known objects. In *BMVC*, pages 1–10. Citeseer, 1995.

[116] F. Stein. Efficient computation of optical flow using the census transform. In CarlEdward Rasmussen, HeinrichH. Bülthoff, Bernhard Schölkopf, and MartinA. Giese, editors, *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 79–86. Springer Berlin Heidelberg, 2004.

[117] C. Strecha, Rik Fransens, and L. Van Gool. Combined depth and outlier estimation in multi-view stereo. In *CVPR*, volume 2, pages 2394–2401, 2006.

[118] C. Strecha, T. Tuytelaars, and L. Van Gool. Dense matching of multiple wide-baseline views. In *ICCV*, pages 1194–1201 vol.2, 2003.

[119] C. Strecha and L. Van Gool. PDE-based multi-view depth estimation. In *Proc. 3D Data Processing Visualization and Transmission*, pages 416–425, 2002.

[120] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, pages 1–8, 2008.

[121] D. Sun, S. Roth, and M.J. Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439. IEEE, 2010.

[122] R. Szeliski. A multi-view approach to motion and stereo. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1. IEEE, 1999.

[123] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live metric 3d reconstruction on mobile phones. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 65–72. IEEE, 2013.

[124] E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, May 2010.

[125] E. Tola, C. Strecha, and P. Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Mach. Vision Appl.*, 23(5):903–920, September 2012.

[126] P.HS Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15(8):591–605, 1997.

[127] W. Trobin, T. Pock, D. Cremers, and H. Bischof. Continuous energy minimization via repeated binary fusion. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, volume 5305 of *Lecture Notes in Computer Science*, pages 677–690. Springer Berlin Heidelberg, 2008.

[128] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008.

[129] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International journal of computer vision*, 59(1):61–85, 2004.

[130] T. Tuytelaars and L.J. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *British Machine Vision Conference*, volume 412, 2000.

[131] R. Tylecek and R. Sara. Depth map fusion with camera position refinement. *Proc. CVWW*, pages 59–66, 2009.

[132] L. Valgaerts, A. Bruhn, and J. Weickert. A variational model for the joint recovery of the fundamental matrix and the optical flow. In *Pattern Recognition*, pages 314–324. Springer, 2008.

[133] J. Van De Weijer and C. Schmid. Coloring local feature extraction. In *Computer Vision–ECCV 2006*, pages 334–348. Springer, 2006.

[134] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[135] H.H. Vu, P. Labatut, J.P. Pons, and R. Keriven. High accuracy and visibility-consistent dense multiview stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):889–901, 2012.

[136] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An improved algorithm for tv-l 1 optical flow. In D. Cremers, B. Rosenhahn, A.L. Yuille, and F.R. Schmidt, editors, *Statistical and Geometrical Approaches to Visual Motion Analysis*, volume 5604 of *Lecture Notes in Computer Science*, pages 23–45. Springer Berlin Heidelberg, 2009.

[137] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An improved algorithm for tv-l 1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45. Springer, 2009.

[138] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber-l1 optical flow. In *BMVC*, pages 1–11, 2009.

[139] Yu-Te Wu, T. Kanade, J. Cohn, and C.-C. Li. Optical flow estimation using wavelet motion model. In *Computer Vision, 1998. Sixth International Conference on*, pages 992–998, 1998.

[140] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *Computer Vision–ECCV 2006*, pages 211–224. Springer, 2006.

[141] J Yao and W.-K. Cham. 3d modeling and rendering from multiple wide-baseline images by match propagation. *Signal Processing: Image Communication*, 21(6):506 – 518, 2006.

[142] M. Ylimaki, J. Kannala, J. Holappa, J. Heikkila, and S.S. Brandt. Robust and accurate multi-view reconstruction by prioritized matching. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2673–2676. IEEE, 2012.

[143] G. Yu and J.-M. Morel. ASIFT: An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line*, 2011, 2011. http://dx.doi.org/10.5201/ipol.2011.my-asift.

[144] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*, pages 214–223. Springer, 2007.

[145] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust tv-l 1 range image integration. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[146] H. Zimmer, A. Bruhn, L. Valgaerts, M. Breuß, J. Weickert, B. Rosenhahn, and H.-P. Seidel. PDE-based anisotropic disparity-driven stereo vision. In *VMV*, 2008.

[147] H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, and H.-P. Seidel. Complementary optic flow. In *Energy minimization methods in computer vision and pattern recognition*, pages 207–220. Springer, 2009.

# Abstract

Image matching is a central research topic in computer vision which has been mainly focused on optical aspects. The aim of the work presented herein consists in the direct use of geometry to complement optical information in the tasks of 2D matching. First, we focus on global methods based on the calculus of variations. In such methods occlusions and sharp features raise difficult challenges. In these scenarios only the contribution of the regularizer accounts for results. Based on a geometric characterization of this behaviour, we formulate a variational matching method that steers grid lines away from problematic regions. While variational methods provide well behaved results, local methods based on match propagation provide results that adapt closely to varying 3D structures although choppy in nature. Therefore, we present a novel method to propagate matches using local information about surface regularity correcting 3D positions along with corresponding 2D matchings.

# Résumé

L'appariement d'images est un sujet central de recherche en vision par ordinateur. La recherche sur cette problématique s'est longuement concentrée sur ses aspects optiques, mais ses aspects géométriques ont reçu beaucoup moins d'attention. Cette thèse promeut l'utilisation de la géométrie pour compléter les informations optique dans les tâches de mise en correspondance d'images. Tout d'abord, nous nous penchons sur les méthodes globales pour lesquelles les occlusions et arêtes vives posent des défis. Dans ces scenarios, le résultat est fortement dépendant de la contribution du terme de régularisation. À l'aide d'une caractérisation géométrique de ce comportement, nous formulons une méthode d'appariement qui dirige les lignes de la grille loin des régions problématiques. Bien que les méthodes variationnelles fournissent des résultats qui se comportent bien en général, les méthodes locales basées sur la propagation de correspondances fournissent des résultats qui s'adaptent mieux à divers structures 3D mais au détriment de la cohérence globale de la surface reconstruite. Par conséquent, nous présentons une nouvelle méthode de propagation guidée par des reconstructions locales de surface.