**ALMA MATER STUDIORUM**
**UNIVERSITÁ DI BOLOGNA**

ARCES – ADVANCED RESEARCH CENTER ON ELECTRONIC SYSTEMS
FOR INFORMATION AND COMMUNICATION TECHNOLOGIES E. DE CASTRO

# RECONSTRUCTION FROM IMAGE CORRESPONDENCES

*Pietro Azzari*

**TUTOR**

*Professor*
***Luigi Di Stefano***

**COORDINATORS**

*Professors*
***Riccardo Rovatti***
***Claudio Fiegna***

PHD. THESIS
*January, 2006 – December, 2008*

PHD PROGRAM IN INFORMATION TECHNOLOGY
CYCLE XXI – ING-INF/05

# Contents

# Abstract

A single picture provides a largely incomplete representation of the scene one is looking at. Usually it reproduces only a limited spatial portion of the scene according to the standpoint and the viewing angle, besides it contains only instantaneous information. Thus very little can be understood on the geometrical structure of the scene, the position and orientation of the observer with respect to it remaining also hard to guess. When multiple views, taken from different positions in space and time, observe the same scene, then a much deeper knowledge is potentially achievable. Understanding inter-views relations enables construction of a collective representation by fusing the information contained in every single image.

Visual reconstruction methods confront with the formidable, and still unanswered, challenge of delivering a comprehensive representation of structure, motion and appearance of a scene from visual information. Multi-view visual reconstruction deals with the inference of relations among multiple views and the exploitation of revealed connections to attain the best possible representation. This thesis investigates novel methods and applications in the field of visual reconstruction from multiple views. Three main threads of research have been pursued: dense geometric reconstruction, camera pose reconstruction, sparse geometric reconstruction of deformable surfaces.

Dense geometric reconstruction aims at delivering the appearance of a scene at every single point. The construction of a large panoramic image from a set of traditional pictures has been extensively studied in the context of image mosaicing techniques. An original algorithm for sequential registration suitable for real-time applications has been conceived. The integration of the algorithm into a visual surveillance system has lead to robust and efficient motion detection with Pan-Tilt-Zoom cameras. Moreover, an evaluation methodology for quantitatively assessing and comparing image mosaicing algorithms has been devised and made available to the community.

Camera pose reconstruction deals with the recovery of the camera trajectory across an image sequence. A novel mosaic-based pose reconstruction algorithm has been conceived that exploit image-mosaics and traditional pose estimation algorithms to deliver more accurate estimates. An innovative markerless vision-based human-machine inter-

face has also been proposed, so as to allow a user to interact with a gaming applications by moving a hand held consumer grade camera in unstructured environments.

Finally, sparse geometric reconstruction refers to the computation of the coarse geometry of an object at few preset points. In this thesis, an innovative shape reconstruction algorithm for deformable objects has been designed. A cooperation with the Solar Impulse project [56] allowed to deploy the algorithm in a very challenging real-world scenario, i.e. the accurate measurements of airplane wings deformations.

# Chapter 1

# Introduction

An individual picture provides a largely incomplete representation of the scene one is looking at. Usually it reproduces only a limited spatial portion of the scene depending on the viewing angle and the position of the observer. The spatial amount of visible scene can be, to some extent, traded with the level of detail; i.e. a full mountain landscape can be grabbed from far away at the cost of missing fine grain details of trees, bushes and skiers, whilst zooomed in snapshots preserve small features but lack mountain peaks and valleys.

The amount of tonal information that can be recorded is severely restricted by the dynamic range of traditional imaging devices, think of a washed out picture of a bright morning light panorama or a dark snapshot of a dimly lit indoor environment. The dynamic range may be adapted to the lighting conditions at hand by configuring exposure settings properly, nonetheless the photometric richess of a real scene greatly exceeds the capability of nowadays CCD sensors.

As the temporal dimension is concerned, only instantaneous information can be recorded, any movement is frozen inside a picture, none can be known about what happens inside the scene immediately after or before the shot is taken. Leaving the shutter open for a while does not usually help since letting the camera integrating over time yields blurred regions where non stationary processes take place.

Moreover, since projective geometry admits many different shapes to exhibit identical projections, very little can be inferred on the 3D geometrical structure of a generic scene from a single view, unless specific prior assumptions are made. Because of that, position and orientation of an observer with respect to the scene remain also hard to guess.

Visual reconstruction methods confront with the formidable, and still unanswered, challenge of delivering a comprehensive representation of structure, motion and appearance of a scene from visual information. Nonetheless, apart for special cases

where single view metrology approaches obtained remarkable results, a comprehensive reconstruction of a given scene is out of reach for single-view algorithms. It is well understood that visual reconstruction approaches relying on multiple views may provide answers to that demanding calls.

## 1.1 Reconstruction from multiple views

The concept underpinning multiple views reconstruction algorithms is the extraction and combination of information coming from several overlapping views, i.e. taken from multiple locations and different instants. When information contained into single views are properly fused together, the collective reconstruction is superior to that possibly attainable by analysing every single image individually.

Visual reconstruction can be ideally split up in many branches depending on the aspect of a scene it aims at retrieving:

- **geometric reconstruction**, it refers to the computation of the 3D structure of a scene. This area can be further subdivided in sparse or dense reconstruction. Sparse reconstruction encompasses a vast number of algorithms knwon as “shape from X”, where X stands for the visual cues employed to perform reconstruction, i.e. motion, shading, defocus, ... These methods usually recover the 3D shape of an object by triangulating rays passing through corresponding points in several calibrated images, namely images whose positions with respect to each other is known. Sparsity refers to the fact the geometric structure is known only at a finite number of points, the structure in between to be inferred with the use of additional constraints, usually reinforcing continuity or smoothness. An example of sparse shape reconstruction is retrieval of a triangulated mesh model of a deformable surface depicted in Fig. 1.1. Conversely, dense reconstruction attempts

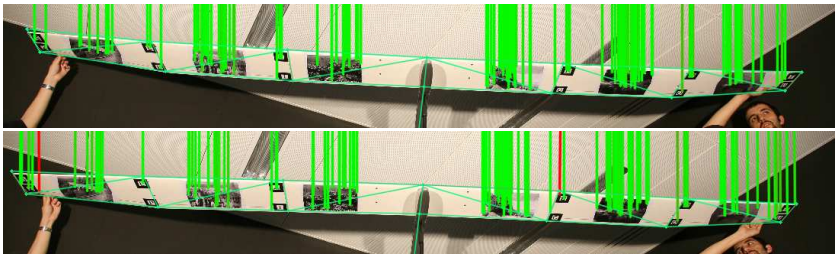


**Figure 1.1:** 2 frames taken from a sequence of 18 portraying an airplane model outlined with the backprojection (green line) of the retrieved 3D structure of its wings.

at delivering the 3D shape at every point in the scene, such description typically

coming along with appearance information such as color or brightness. It is usually deployed in simpler geometric scenarios and some representative family of algorithms are image mosaicing and two-view dense stereo. Intuitively, a dense reconstruction of the appearance of a scene may be obtained by combining several images taken from different viewpoints and properly stitched together so that common parts of the scene overlap in the final picture, usually dubbed mosaic (see Fig. 1.2).

**Figure 1.2:** (Top) 8 frames from a sequence of 680 captured by a hand-held camera. (Bottom) All 680 frames combined in a mosaics with much greater field of view.

- **photometric reconstruction**, it aims at recovering the photometric content of a scene that might be lost due to limited dynamic range of the imaging device or unfavorable lighting conditions. Along the same line, reconstruction takes place by composing many snapshots taken with variable exposure settings, each of them capturing a different range of radiance. The combination of several overlapping ranges allow to extend the collective dynamic range, leading to images with typical range resolution of 16 or 24 bits per channel.

- **camera pose reconstruction** is usually referred to as calibration and sometimes is included in sparse geometric reconstruction algorithms as an early phase. It addresses the problem of recovering the relative position and orientation of set of images with respect to a given coordinate frame. For example, the reconstruction of the motion trajectory of an object throughout a sequence can be accomplished by comparing a reference view of the object in its rest position with all the frames of the sequence. The displacements of corresponding structures across the sequence hint at the trajectory the camera has followed. The knowl-

edge of the position of the camera with respect to the scene triggers a variety of applications; among the most popular stands Augmented Reality (AR). An example of Augmented Reality is displayed in Fig. 1.3) where 3 virtual objects are realistically rendered as they were laying on top of the showcase counter according to the reconstructed pose of the camera.

**Figure 1.3:** 3 frames from a sequence of 420 captured by a hand held camera. The accurate retrieval of camera motion allows the virtual objects to realistically lay at the same places in every frame.

Every visual reconstruction algorithm, irrespective of the class it belongs to, requires some kind of relations to be established among the set of analyzed images. In order to accomplish this task two key steps are invariably present: image matching and image registration. The former step refers to the detection and matching of salient features (points, areas or structures) among images. The identification of corresponding features in multiple views hints at the presence of spatial, tonal or temporal relation among the set of images. Image registration is concerned with the quantitative computation of the inter-images relations given a set of corresponding salient features. Both image matching and registration are very active fields of research and some of the most relevant achievements will be discussed respectively in chapter 2.1 and chapter 2.2.

This thesis investigates on novel methods and applications in the field of reconstruction from multiple views. Three main threads directed the investigation: dense geometric reconstruction in the context of image mosaicing and its applications, motion trajectory recovery applied to mixed reality and vision-based human-machine interfaces, sparse structure and motion reconstruction for deformable surfaces. All the algorithms conceived have been tested on both synthetic and real image sequences, and data sets have been made available for researchers active in the same field.

## 1.2 Fields of application

Visual reconstruction encompasses a wide number of concepts, ideas and algorithms enabling established as well as emerging technologies and applications. Dense geometric reconstruction, in the form of image mosaicing, has already made an impact

into the digital photography market with the continual release of new products which allow a handful of photos or even video stream from an hand-held camera to be stitched together into a wide field mosaic. Interactive 360 mosaic is routinely used to illustrate and promote holiday resorts, museums, historical and archeological sites. Recently image mosaics have found application in visual surveillance systems that deliver motion detection using pan-tilt-zoom cameras, as discussed in chapter 3. On the other hand, dense two-view stereo reconstruction has become a cornerstone for robust navigation of unmanned vehicle and robots and will probably hit the market soon.

Sparse geometric reconstruction is making its way inside commercial software for vision-based shape computation tailored for architect and engineers studios. On a more precompetitive stage of development, a pair of applications addressing non contact shape retrieval of complex deformable surfaces in uncontrolled environment such as airplane wings and boat sails are described in chapter 6.

Camera motion reconstruction has become a valuable tool for visual effects technology such as match moving, namely the insertion of virtual objects into real footage. Automatic computation of the correct position, scale, orientation and motion in relation to the photographed objects in the scene greatly simplifies and speed up match moving tasks. The same functionality has found useful applications in mixed or augmented reality, see 5.1, and human machine interface, refer to 5.2 for gaming related applications. Another fertile field of application is automatic steering, landing and docking of unmanned vehicles.

## 1.3 Structure of the thesis

This thesis is subdivided into three main parts focusing on different aspects of visual reconstruction. As mentioned before, any multiple views visual reconstruction approach builds on top of two pillars: an image matching method to infer the relations among the set of images and image registration algorithms to numerically appraise them. For this reason, chapter 2 is devoted to the presentation of principles and algorithms dealing with image matching and image registration.

As far as research activities are concerned, dense structure reconstruction has been investigated first during this thesis work. In particular the focus has been on the demanding problem of reconstructing the appearance of a scene through image mosaicing in the context of visual surveillance. Basically, given several shots taken by a pan-tilt-zoom (PTZ) camera, a mosaicing algorithm aims at the generation of a unique image of higher resolution and field of view, called mosaic. A visual surveillance based on a PTZ camera can then use a mosaic as reference image (i.e. background) so as to rely on standard and well established motion detection techniques developed for the static

camera scenario. Other than the unknown motion of the camera, other difficulties such as changes in lighting, exposure, independently moving objects and optical distortions compete to render this problem a hard one. Moreover, the use in the context of visual surveillance imposes real-time computation requirements. In chapter 3 an original algorithm for real-time image mosaicing is detailed with validation tests accomplished on real image sequences taken by a PTZ surveillance cameras.

It soon became clear that both visual inspection and other statistical measures, such as residual fitting error, were not discriminant nor reliable indicators of the quality of a mosaicing algorithm. Nonetheless, to the best of our knowledge, no established or widely employed data sets, performance metrics or evaluation methodologies have been proposed in literature to quantitatively appraise the performance of mosaicing algorithms. Such a shortage is very detrimental to the development of this research field, for it hinders the objective assessment and comparison of different proposals meanwhile complicating communications and collaborations efforts among researchers. Chapter 4 addresses this issue and describes a proposal of an evaluation methodology for image mosaicing algorithms comprehensive of standard data sets, performance metrics and comparison procedure. The methodology has been made available to the scientific community through a publicly accessible website.

Camera pose reconstruction has been the second field of investigation. This topic is concerned with the determination of the position and the orientation of a camera with respect to a given scene. When a scene or parts of it can be assumed flat, several theoretical analogies arise with image mosaicing techniques whereas in place of the appearance of the scene the focus is on the position of the cameras observing it. In this context, two applications that would greatly benefit from automatic pose estimation have been examinated: Augmented Reality and Human-Machine Interfaces (HMI). In chapter 5.1 an original use of mosaics in a AR context is proposed; the point is to show how image mosaicing can boost the performance of established planar pose estimation algorithms. Chapter 5.2 deals with the introduction of vision-based pose estimation in the field of interfaces for gaming applications. Two videogames, built on top of the camera-based interface have been developed .

The third and last research direction has been sparse geometric reconstruction of deformable objects. Here the scope is to estimate a low-dimensional geometrical representation, for instance a triangulated mesh, of the 3D structure of a flexible surface such as journals, cloths, flags and so on. While the piecewise planar assumption about the structure of the object is usually a reasonable approximation in this case too, the capability of the object to deform introduces a whole new family of projection ambiguities. While the theoretical implications have been extensively studied, real-world demonstrations have been much less compelling being limited only to reconstruction

of sheet of papers and napkins. Chapter 6 reports on a novel vision-based method for measuring airplane wings deformations using a single camera. Both synthetic and real images have been employed to assess the performance of the conceived algorithm.

The last chapter 7 summarizes achievementes and lessons, draws conclusions and traces future directions and foreseeable developments and advances.

## 1.4 Summary of contributions

The principal contributions and the scientific results, in terms of peer-reviewed publications on international conferences and journals and unpublished tech reports, originated from the research activities carried out during the PhD course is as follows:

**Chapter 3: Real-time mosaicing for visual surveillance**

- An original near real-time registration algorithm for the construction of globally coherent image mosaics apt to detect motion in visual surveillance systems.
- A fast and exact histogram specification algorithm for handling photometric registration of differently exposed images during the construction of image mosaics.

1. P. Azzari. General purpose real-time image mosaicing. In *Proc. of ICVSS 2007*, July 2007.
2. A. Bevilacqua and P. Azzari. A high performance exact histogram specification algorithm. In *Proc. of ICIAP 2007*, pages 501-512, September 2007.
3. A. Bevilacqua and P. Azzari. A fast and reliable image mosaicing technique with application to wide area motion detection. In *Proc. of ICIAR 2007*, pages 501-512, August 2007.
4. A. Bevilacqua and P. Azzari. High-quality real time motion detection using PTZ cameras. In *Proc. of Intl. Conf. on AVSS 2006*, pages 23, November 2006.
5. P. Azzari and A. Bevilacqua. Joint spatial and tonal mosaic alignment for motion detection with PTZ camera. In *Proc. of ICIAR 2006*, pages 764-775, September 2006 (oral).

**Chapter 4: Evaluation methodology for image mosaicing algorithms**

- A comprehensive evaluation methodology for image mosaicing algorithms designed to objectively compare and rank approaches within a busy and, until then, inordinate research field. Evaluation procedures and data sets have been released for public use through freely accesible webpages.

1. P. Azzari, L. Di Stefano, S. Mattoccia. An evaluation methodology for image mosaicing algorithms. In *Proc. of Intl. Conf. on ACIVS 2008*, pages. , October 2008 (oral).

**Chapter 5: Camera pose reconstruction and its applications**

- Original usage of image mosaics for the enhancement of accuracy and steadiness of pose estimation algorithms. The approach has been successfully integrated into an existing augmented reality system aimed at aereonautical maintenaince.
- An innovative vision-based interface for videogames designed for easier and more pleasant gaming experience.
- Two original gaming applications built on top of the interface have been developed.

1. P. Azzari, L. Di Stefano. Vision-based markerless gaming interface. In *Proc. of Intl. Conf. on Image Analysis and Processing*, 2009 (submitted).
2. P. Azzari, Robust image registration using linear and quadratic programming. Tech report, CV Lab, University of Bologna, Italy, 2008.
3. P. Azzari, Image registration using SVM regression. Tech report, CV Lab, University of Bologna, Italy, 2008.
4. P. Azzari, L. Di Stefano, F. Tombari, S. Mattoccia. Markerless augmented reality using image mosaics. In *Proc. of ICISP 2008*, pages , July 2008 (oral).

**Chapter 6: 3D reconstruction of deformable surfaces**

- Thorough design and test of a monocular measurement system for wing deformations. Full and precise 3D reconstruction of the shape is delivered regardless of the position or deformations of the analyzed surface.

1. K. Startchev, P. Azzari, P. Lagger, A. Varol, and P. Fua. Video-based measurements of deformable surfaces. In *Journal of Machine Vision and Applications*, (in preparation).
2. P. Azzari, P. Fua and P. Lagger, Video-based measurements of wing deformations. Tech report, CV Lab, Ecole Polytechnique Federal Lausanne, Switzerland, 2008.

# Chapter 2

# Theoretical background

A reasoning process dealing with more than one view requires firstly to reveal and quantify the relationships subsisting among the set of images at hand. Visual reconstruction from multiple view algorithms make no exception, for they always build on top of reliable image matching and registration techniques. Since these techniques are essential and unfailing, the present chapter is devoted to illustrate the concepts and algorithms mostly recurring in the remainder of the thesis.

## 2.1 Image matching with keypoint correspondences

Image matching is a research area mainly concerned with the discovery of connections among a set of images. In its wider meaning, the nature of such connections could refer to relationships as diverse as geometric, photometric, temporal and so on. For example, a pair of partially overlapping images could be surely cast in some kind of geometric relationships for they are both observing the same scene probably from slight different viewpoint or with different cameras. If the latter is the case, photometric relationships among corresponding pixels could probably hold since different cameras usually have different responses to incoming radiance. Moreover, temporal relations can be revealed when dynamic events are observed in multiple images, for instance, the amount of daylight could hint at the time and the order pictures have been taken.

Although several methods have been conceived to reveal inter-image connections the concept of salient features extraction is widespread. Feature extraction is most of the time inevitable since using an entire image as an observation is difficult or impossible due the high dimensionality (typically the order of a hundred thousand pixels). Nonetheless salient features could be anything ranging from points, lines, curves to textures, image structures, blobs and so on.

In this section only keypoints-based image matching algorithms are treated. The ultimate goal of such class of algorithms is to deliver a set of image point correspondences $x_i \leftrightarrow x_i^{'}$, where $x_i$ in $\mathbb{R}^2$ are the keypoints detected in one image and $x_i^{'}$ in $\mathbb{R}^2$ are those detected in a second image.

Three performance figures are important for image matching algorithms based on keypoints:

- **repeatability** refers to the ability to select the same points of a scene in different images independently of the changes in viewpoint, lighting, scale and so on.

- **distinctiveness** is concerned with the discriminative power of the description; different points should always exhibit very diverse descriptors so that mismatch probability is minimized.

- **accuracy** pertains to the precise localization of a keypoints inside an image, subpixel methods have become common in order to increase performance.

Ideally, the best image matching algorithm is the one that find discriminative descriptors that can be matched with high reliability and accuracy between frames, while also finding a large number of features per frame.

Inside the class of keypoints-based image matching algorithms the focus will be on the three most popular and representative feature extractors used nowadays in computer vision: the Harris corner detector, the Kanade-Lucas-Tomasi tracker (KLT), and the Scale-Invariant Feature Transform (SIFT). The term feature extractor is used to describe the combination of a feature detector, or keypoints detector, and a feature descriptor. Detectors are used to find keypoints in an image, after which a descriptor is created that describes the local neighborhood around the points. An overview of the state of the art in feature extractors is given by Mikolajczyk and Schmid [10]. A feature tracker establish correspondence among keypoints detected in different images by comparing their respective descriptors. A keypoints-based image matching algorithm is the ensemble of a feature extractor and a feature tracker.

### 2.1.1 Harris corner detector

The Harris corner detector, named after the authors that presented it in the first place [5], is one of the most widely used and established keypoint detectors. Harris keypoints or corners, sometimes also referred to as interest points, are image features characterized by high intensity changes in two orthogonal directions. For instance, if a square object is present in the image then its four corners are usually very good interest points.

A formal statement of corners requires the introduction of the Harris local structure

matrix $C$ which is defined as

$$C = w_G(\sigma) * \begin{bmatrix} \sum\sum_R \left(\frac{\partial I}{\partial x}\right)^2 & \sum\sum_R \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \sum\sum_R \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \sum\sum_R \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} \tag{2.1}$$

where $I$ is the image at hand and $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ its partial derivatives, $R$ is a $(2\times d+1)\times(2\times d+1)$ neighboring image region around $(x,y)$, $w_G(\sigma)$ is a Gaussian kernel with standard deviation $\sigma$ and * denotes convolution.

Let $\lambda_1 \geq \lambda_2$ be the two eigenvalues of the matrix $C$. Since $C$ is symmetric and positive semi-definite, both $\lambda_1$ and $\lambda_2$ are non-negative. The values of these eigenvalues directly admit some useful interpretations:

- in a uniform and homogeneous region, $\lambda_1 = \lambda_2 = 0$.
- at the location of a step edge, $\lambda_1 > \lambda_2 = 0$. The corresponding eigenvector for $\lambda_1$ is associated with the direction that is orthogonal to the edge.
- at the location of a corner, $\lambda_1 \geq \lambda_2 > 0$. The larger are the values of $\lambda_1$ and $\lambda_2$, the higher are the contrasts of the edges orthogonal to the directions of the corresponding eigenvectors.

Given the previous definition, the Harris corner detector proceeds as follows:

1. for each image point $(x,y)$:
   - construct the local structure matrix $(x,y)$
   - compute the response to the "cornerness" filter $r$ defined at each pixel coordinates $(x,y)$ defined as
   $$r(x,y) = det(C(x,y)) - k(trace(C(x,y))^2; \tag{2.2}$$
   where $k$ is an adjustable constant.
2. perform a non-maximal suppression on the "cornerness" filter $r$ response to suppress weak corners around the stronger ones.
3. threshold the residual response according to a threshold value $t$.

Altogether, the Harris corner detector requires three additional parameters to be specified: the constant $k$, the radius $d$, of the neighbourhood region for suppressing weak corners, and the threshold value $t$. Different configurations of such parameters may yield very diverse outcomes, nonetheless this is out of the scope of this section, for further investigation please refer to the original work [5].

The descriptor associated to each detected corner is just the image intensity neighborhood around the interest point. The matching phase is accomplished by comparing the descriptors using the $L^2$ norm, a low score, originated by similar image patches, signaling probably correspondent pair of corners.

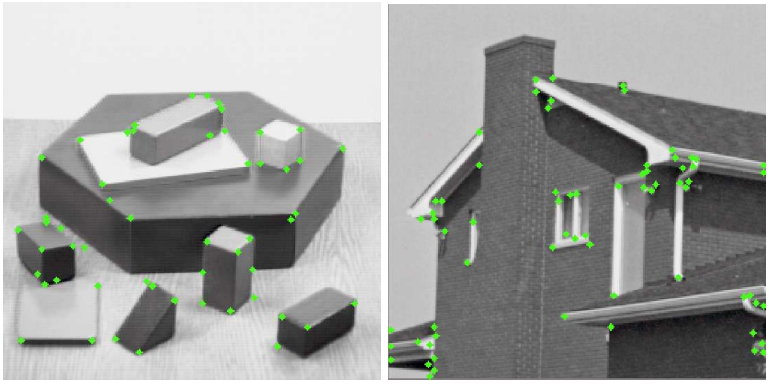


**Figure 2.1:** Two test images showing the keypoints detected by the Harris detector.

### 2.1.2 Lucas-Kanade-Tomasi feature tracker

The Kanade-Lucas-Tomasi (KLT) corner detector [9] is almost contemporary to Harris proposal and shares many concepts with it. For instance, the KLT detector relies on the local structure matrix $C$ defined in Eq. 2.1

The KLT feature detector consists of these steps:

1. for each image point $(x, y)$:
   - construct the local structure matrix $C$ around $(x, y)$.
   - compute the smallest eigenvalue, $\lambda_2$, of the matrix $C_{KLT}$ $(x, y)$;
   - if $\lambda_2 > \lambda_{min}$, save $(x, y)$ into a potential corner list, $L$.
2. sort $L$ in decreasing order of $\lambda_2$
3. scan the sorted list from top to bottom and select points in the list in sequence.

Points that fall inside the neighborhood R of any selected points are removed. The output produced by the KLT corner detector is a list of corner points that have $\lambda_2 > \lambda_{min}$ and the neighborhood $R$ of these points do not overlap. Similarly to Harris detector, the KLT algorithm admits two parameters:

- threshold value, $\lambda_{min}$, on the second eigenvalue $\lambda_2$, and
- a neighborhood window radius $d$.

Indeed, results are very similar to the Harris technique, as may be noticed by comparing figures 2.1 and 2.2.

Like Harris, the KLT descriptor consists of a neighboring image patch and point correspondences are established according to the correlation score among patches. A

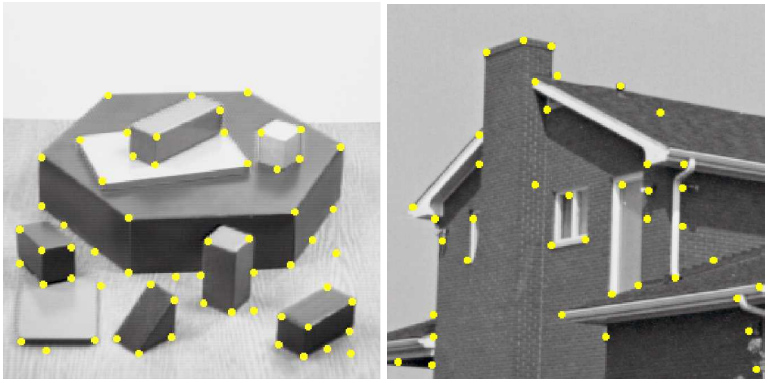


**Figure 2.2:** Two test images showing the keypoints detected by the KLT detector.

novel matching algorithm, introduced later by Lucas and Kanade, uses a gradient descent method to iteratively align image intensity patches using an affine warping model [2, 11].

### 2.1.3 Scale Invariant Feature Transform

The SIFT (Scale Invariant Feature Transform) keypoint detector/descriptor was proposed by Lowe in 1999 [7, 8]. The SIFT features are feature vectors that represent local image measurements, which have been reported to be invariant to image translation, scaling and rotation and partially invariant to changes in illumination and local image deformations.

The SIFT detector locates keypoints as follows (see Figure 2.3):

- the input image, $I(x,y)$, is convolved with a number of Gaussian filters whose standard deviations $\{\sigma_1, \sigma_2, ...\}$ differ by a fixed scale factor. The convolutions yield a small number of smoothed images, denoted by $\{G_{\sigma_1}(x,y), G_{\sigma_2}(x,y); ...\}$
- adjacent smoothed images are pairwise subtracted to yield DoG (Difference-of-Gaussian) images, according to

$$D_{\sigma_j}(x,y) = G_{\sigma_{j+1}}(x,y) - G_{\sigma_j}(x,y) \tag{2.3}$$

- smoothed images from Step 1 are subsampled and the procedure in Step 2 is repeated on the subsampled images, yielding a number of DoG images over the scale space.
- each point in these DoG images is examined. A keypoint is marked at a location where the point is a local minimum or maximum of its 8 neighbours on the same scale and of its 9 neighbours on the scales above and below.

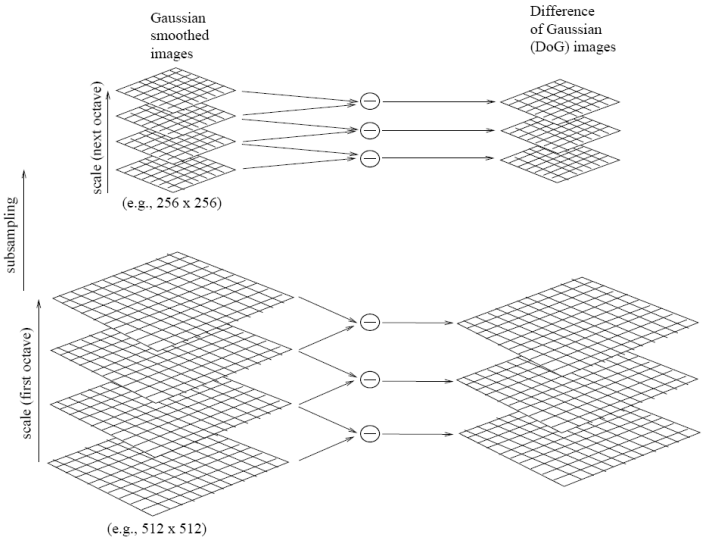



**Figure 2.3:** The DoG image pyramid used by the SIFT detector to locate keypoints.

The surrounding intensity and gradient information around each keypoint are encoded in the SIFT descriptor. The neighborhood region around the keypoint is subdivided in a regular grid of $4 \times 4$ cells. Image gradients are computed within each cell and classified into 8 orientations (see Fig.2.4), giving a SIFT descriptor of 128 elements long for each keypoint.

Unlike Harris and KLT, SIFT keypoints are not always located at corner points as may be noticed in fig. 2.5. Nonetheless, SIFT keypoints have shown high repeatability and distinctiveness in some of the most challenging computer vision applications such as wide-baseline stereo and multi-view reconstruction.

The matching phase is accomplished by computing the euclidean distance between normalized feature descriptors, with the addition constraint that the nearest neighbor must be sufficiently closer than the second closest neighbor. The idea stems from the observation that false matches caused by noise ought to have multiple noisy matches at similar distances [8].

## 2.2 Planar image registration

A pair of corresponding image points $x_i \leftrightarrow x_i^{'}$ are projections in two images of the same pre-image point $X_i$. A set of corresponding image points $x_i \leftrightarrow x_i^{'}$ for $i = 1, 2, ..., n$, detected in a pair of images, hints at the fact that the views are related to some extent. The explicitation and quantification of the subsisting relations is demanded to image

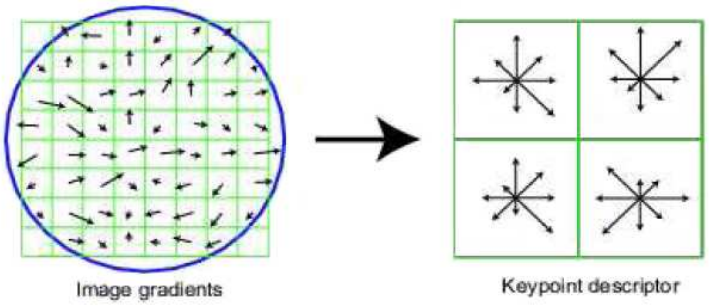



**Figure 2.4:** This figure shows a simplified example of a $2 \times 2$ descriptor array computed from an $8 \times 8$ window of image gradient vectors. The SIFT detector reported in [8] works on $16 \times 16$ windows of image gradient vectors, giving descriptors of 128 elements in length.

registration algorithms. In this section a number of concepts regarding the geometry of two views are treated, in particular registration of images of planar structures is emphasized, for it is a useful approximation in many circumstances and oftentimes used throughout the thesis. The extension to an a arbitrary number of views has been treated by iteratively applying two views algorithms to a shifting pair of images.

Hereinafter, it is assumed that an image point $x = [u, v]$ is projection of a 3D space point $X = [X, Y, Z]$ imaged by the camera according to the perspective projection matrix $P$:

$$s\widetilde{x} = P\widetilde{X} = K \begin{bmatrix} R & t \end{bmatrix} \widetilde{X} \text{ with } K = \begin{bmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

where $\widetilde{x} = [u, v, 1]$ and $\widetilde{X} = [X, Y, Z, 1]$ are the homogeneous representation of $x$ and $X$ respectively. In Eq. 2.4 $s$ is an arbitrary scale factor; $(R, t)$, called the extrinsic parameters, is the rotation and translation which relates the world coordinate system to the camera coordinate system; $K$ is called the camera internal matrix, with $(u_0, v_0)$ the coordinates of the principal point, $\alpha$ and $\beta$ the scale factors in the $u$ and $v$ axes, $c$ the parameter describing the skewness of the two image axes.

In the general case of an arbitrary scene observed by two views, characterized by projection matrices $P$ and $P'$, a corresponding image pair $x \leftrightarrow x'$) is linked by the fundamental matrix $F$

$$\widetilde{x}'^T F \widetilde{x} = 0 \tag{2.5}$$

Since a valid fundamental matrix is a $3 \times 3$ matrix of with rank 2, any image points in one image is put in correspondence with a line in the second image, depending on the 3D structure of the imaged scene. Such ambiguity cannot be solved from image

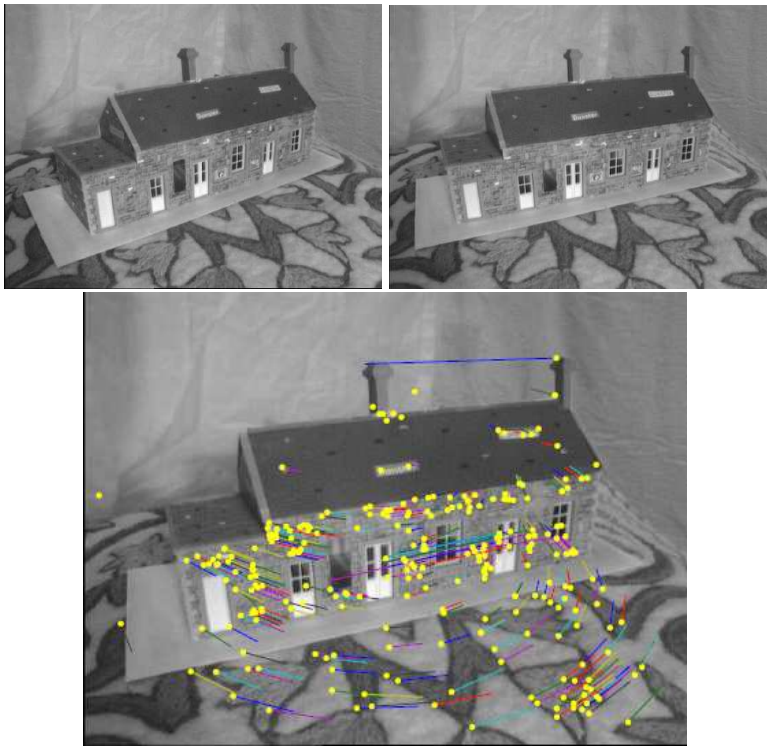


**Figure 2.5:** (Top) Two test images taken from slightly rotated viewpoints. (Bottom) Detected SIFT keypoints inside the right image are marked with yellow dots. Colored lines connects keypoints found in the right image and their corresponding locations in the left image. Three cases are shown: unmatched keypoints (i.e. on the middle left), mismatched keypoints (on the pair of chimney-pots) and correct matches.

correspondences alone and gives raise to a whole family of valid projection matrices $P$ and $P^{'}$ satisfying Eq. 2.5. Hence little can be inferred unless prior assumption are made either on relative position of the cameras or structure or the scene.

Assuming the observed scene, or part of it, has a planar structure greatly simplifies theory and calculations. Even though it may seem a strong approximation, the planarity assumption is acceptable in many scenarios and has been widely applied. As far as this thesis is concerned such a simplification holds for:

- **dense structure reconstruction through image mosaicing**. Since this topic is mainly concerned with the creation of a wide angle image, such as in panorama photography or in wide area surveillance, the presence of almost flat scenes, i.e. scene in which relative depth is negligible with respect to the distance from the camera, is quite common. Moreover, the case of purely rotating cameras, for instance PTZ cameras, observing arbitrary scenes, is governed by the same

geometry relations as in the former case.

- **pose reconstruction**. A large number of objects in real life are flat or contains flat parts. This is all the more true when thinking at objects as being piecewise flat, as a polyhedral mesh of small polygons joint together. The smaller the polygons the more precise the approximation.

- **sparse structure reconstruction of deformable objects**. As before objects can be thought of being composed of flat parts connected by joints that let them flex. The piecewise flat model, for example a triangulated mesh, holds even for many deformable objects. Additional smoothness constraints are needed to handle deformation degrees of freedom properly.

Planar image registration, a subset of the multiple view registration area, applies to the cases where geometric relations link views, or part of them, that portray flat regions. The most general type of relation among keypoints pairs $x \leftrightarrow x'$) belonging to corresponding flat regions in different images is modeled by a homographic relation as

$$s\widetilde{x} = H\widetilde{x'} \tag{2.6}$$

where $H$ is a $3 \times 3$ matrix of rank 3, called homography, defined as

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \tag{2.7}$$

It may be shown that the geometric relations in any of the abovementioned cases can be cast into an homography estimation problem. Due to its vast field of applications many algorithms have been conceived for computing a homography given a set of correspondence pairs, for hinstance homogeneous and inhomogeneous DLT, Sampson approximation and so on, as illustrated in [6]. Every method differs from each other for the criterium used for the estimation of $H$, namely the type of distance $d$ to be minimized. Nonetheless, all the algorithms perform a Least Square (LS) minimization of an error $e$ that may be expressed in the following form

$$e = \sum_{i=0}^{N} d(x_i, x_i')^2 \tag{2.8}$$

where $d$ can be in principle any linear or non linear function of the unknown entries of matrix $H$.

As a final remark, it is worth pointing out that standard LS methods are very sensitive to data, i.e. corresponding point coordinates, affected by non gaussian noise, hereinafter “outliers”. For example, inaccurate location of corresponding points or

false matching are usually randomly distributed and may heavily affect the solution dragging the LS estimation of $H$ well away from the true one. Two main approaches may be adopted to estimate parameters of a mathematical model from a set of observed data which contains outliers. The first approach adopt an explicit filtering stage to sift data before the LS estimation, for example RANSAC (RANdom SAmple Consensus) [4] is a popular outlier removal approach. The second approach relies on the use of statistically robust distance functions. Both approaches have been successfully applied, the choice depending on on the context and the noise presumably affecting the data, for further investigation refer to [6, 1].

Throughout the thesis, an enhanced version of the original RANSAC algorithm, suggested in [3], is used. The former method considers many random data subsets, each containing the minimum number of samples required to compute the model parameters exactly, and select the parameter set which has the largest number of compatible data. Eventually the model parameters are refined using an as large amount of data as possible, namely every compatible point correspondence.

The innovative part consists in iterating the process by using the estimated homography to bootstrap a new search for point correspondences. The search procedure proceeds as follows: given an interest point $x_i$ in the first image, a match is sought in a search window centered on the expected position $x_i = Hx_i^i$ in the second image ($H$ is the identity matrix at first iteration). Because the search is now guided, there is a probability of fixing false matches established at the previous step augmenting the total number of valid correspondences. The new set of inliers is again used to refine the estimate of H. The estimation and guided matching stages are repeated until the number of valid correspodences stabilizes.

# Bibliography


[1] P. Azzari. Robust image registration using linear and quadratic programming. Technical report, CV Lab Tech Report, University of Bologna, Italy, 2008.

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

[3] D. P. Capel. *Image mosaicing and super-resolution*. University of Oxford, 2001.

[4] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381395, 1981.

[5] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of Alvey Vision Conf,*, pages 147–151, 1988.

[6] R. Hartley and A. Zisserman. *Multiple view Geometry in computer vision*. Cambridge University Press, Second Edition, 2003.

[7] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of Intl. Conf. on Computer Vision*, pages 147–151, 1988.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60(2):91–110, 2004.

[9] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of Intl. Joint Conf. on Artificial Intelligence*, pages 674–679, April 1981.

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

[11] C. Tomasi and J. Shi. Good features to track. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

# Chapter 3

# Real-time image mosaicing

Image mosaicing is stirring up a lot of interests in the research community either for its scientific significance as well as for the potential implications in real world applications; indeed, automatic image alignment and stitching is key to several higher level image processing tasks.

Next section presents a real-time mosaicing algorithm capable of constructing high quality seams-free panoramic images. The proposed algorithm performs a fully automatic spatial and tonal registration by exploiting keypoints correspondences and histogram matching techniques. Remarkably, the approach does not rely on a priori assumption, with all the required information extracted from the image set. A rich set of image sequences has been collected to test the algorithm and assess its stability and flexibility. In addition, the approach has been successfully integrated in a visual surveillance system in which the mosaic is used as background to perform motion detection and tracking with a Pan Tilt Zoom (PTZ) camera.

The second next section investigates further on the problem of accurate tonal alignment of a set of spatially registered images. Aside being a key element of any image mosaicing algorithm, tonal alignment can be regarded as a stand alone topic as long as principled handling and processing of differently exposed images calls for photometric normalization.

## 3.1 On-line image mosaicing for visual surveillance

Image mosaicing is a popular method for effectively increasing the field of view of a camera by allowing several views of the same scene to be combined into a single image, called a mosaic. Stitching together multiple images taken from different viewpoints allows to create large field-of-view pictures, up to 360 degrees, with consumer-grade

camera and without introducing the undesirable lens deformation usually accompanying wide-angle lenses.

Properly handling multiple images taken at different locations, instants and light conditions requires mosaicing methods to be robust with respect to viewpoint and illumination changes, scene multimodality (i.e. waving trees and hedges), moving objects, imaging device noise and so on. Other camera related aspects such as varying intrinsics (focal length, principal point location) should not degrade significantly the performance of the system. Finally, inherently real time applications, such as visual surveillance, require the method to perform on line at acquisition rate. Ensuring consistent geometric and photometric reconstruction of the scene by continuously combining pictures in real-time in a mosaic is a challenging goal.

The next sections describe a real-time image mosaicing technique devised to construct high quality mosaics from video sequences offering all the above mentioned desired properties. Spatial and tonal consistence is achieved by exploiting an original dual geometric registration scheme, illustrated in section 3.1.2 , and a fast photometric registration stage, introduced in section 3.1.2 and further detailed in section [4]. Furthermore, the algorithm has been conceived to be completely *image based*. No prior information, such as camera calibration (focal length, distortion coefficients), scene geometry or feedback signals coming from the imaging device (pan/tilt angular movements, exposure settings), is necessary for the mosaic to be built. Instead all the required information is automatically extracted from the images, yielding a hardware independent and general purpose algorithm.

The quality of the attained mosaics has been initially verified by visual inspection. Although human perceived quality is a largely subjective indicator, it seemed to be correct since only qualitative applications such as digital photography, photomontage and so on were initially envisioned.

However, as we addressed the adoption of mosaics within the visual surveillance domain, we begun to rely on overall system performance as a quantitative quality metric. The idea of mosaic-based motion detection connects with the attempts to improve motion detection systems developed by researchers in the last years. Some proposed solutions involved the use of PTZ cameras to widen the surveyed area. Despite of the many available methods for background difference, none of them can be trivially extended to work with hinged PTZ cameras. Mosaic-based motion detection consists in the use of traditional background subtraction techniques on a panoramic background image built by means of a mosaicing algorithm.

This section is composed as follows. Section 3.1.1 provides an overview of the state of the art in the image mosaicing research field. Section 3.1.2 describes the proposed algorithm for real-time image mosaicing, detailing both spatial and tonal alignment

phases along with image blending and warping steps. Mosaic-based motion detection, using moving cameras is examined in Section 3.1.3. Experimental results on real sequences are reported in Section 3.1.4, followed by concluding remarks and possible directions for future research on this topic in Section 3.1.5.

### 3.1.1 Previous work

**Image mosaicing algorithms**

During the last decades a considerable number of scientific works addressed the broad topic of image registration (for a comprehensive survey refer to [45]). In their diversity, published methods share a lot of theoretical and technical aspects. According to the adopted image matching techniques, the algorithms can be coarsely classified into two main families: direct methods and feature-based methods. As long as the number of frames simultaneously combined is considered, two further categories may be recognized: sequential methods and global registration methods. Inside these super classes, methods can be further distinguished according to the preferred geometrical and photometric model, treatment of independently moving objects, self-calibration capabilities and so on. A first simple dichotomic taxonomy is proposed here to highlight the two main different image matching approaches:

- Direct methods. These algorithms usually attempt at iteratively estimating the transformation parameters by minimizing an error function based on pixelwise brightness differences in overlapping areas [11, 39, 41, 26, 36, 42]. The advantage of direct methods is highly accurate registration and reconstruction, due to the exploitation of information associated with every single pixel. Image formation process non idealities, such as illumination changes [11, 26], lens distortions [36, 42] and vignetting, can be accounted for in the pixelwise error function. The main drawback of this class of methods is the high computational cost due to the non-linear parametrization of the error functions, which call usually for complex iterative algorithms. Moreover, an initial guess for the parameters is required to avoid local minima. Since direct methods are usually incompatible with real time constraints, they often find application in batch registration processes, where maximum accuracy is the goal. Furthermore these algorithms are sensitive to moving objects in the scene and their presence may cause serious performance degradation [11, 39, 42].

- Feature-based methods. As opposed to using all the available pixels, these methods establish feature correspondences among images to be registered [5, 44, 9, 1, 12, 17]. Many different features have been used in the literature, including

regions, lines and keypoints. Recently, keypoints gathered a large agreement among researchers, becoming the de facto standard for image registration purposes ([31, 40, 3]). After being detected in one image, keypoints are searched in a second image based on descriptor similarity. Unlike direct methods, feature based methods admit linear error functions and hence solutions may be found in closed form. By selecting appropriate features [9, 17], these methods can be very robust to illumination changes, image rotations and zooming. Furthermore, moving objects in the scene are tolerate as long as appropriate filtering schemes, such as RANSAC [5, 1], or robust error functions, are deployed to deal with incompatible keypoints.

As far as the registration problem is concerned, two classes of algorithms may be distinguished as well:

- Global registration methods compute the best alignment among several images by simultaneously minimizing the misregistration between all the overlapping pairs of images [11, 39, 41, 26, 36, 42, 9, 12, 17]. Global registration algorithms deliver the most consistent geometric reconstruction and have been proposed in conjunction with both direct [11, 39, 36, 42] and feature-based approaches [9, 12, 17]. The joint optimization is usually computationally intensive, moreover these methods require all the images to be known in advance. Any update to the image set requires the computation to start over again, hence ruling out even the possibility of performing on-line, although slowly.

- Sequential algorithms allow the construction of a mosaic by continuously combining new images as soon as they become available. Every new image is aligned with the previous one (Frame to Frame registration) or with the mosaic built thus far (Frame to Mosaic registration). Intuitively, alignment of pair of images is simpler a problem than multiple view alignment, thus yielding a faster computation that holds the potential for real-time operation. Moreover, these algorithms can usually handle an indefinitely high number of images and does not need to know all images in advance. Nonetheless, pairwise registration is only locally optimal since past frames are not explicitly taken into account, moreover the sequential combination of images may lead to visual artifacts due to error accumulation. The next section hosts a quick overview of sequential mosaicing algorithms, usually devised in combination with real time applications, such as motion detection.

**Sequential mosaicing for motion detection**

Motion segmentation of video sequences is widely recognized as being the first layer of many video processing applications such as video urveillance, traffic monitoring and human activity understanding. Among the possible ways for detecting motion, background subtraction can provide the most accurate segmentation of moving objects, but requires the use of a stationary camera. Moving masks are extracted by thresholding the absolute difference between a reference image (referred to as background) and the current frame. Also, background maintenance activities are envisioned in order to keep the background up-to-date in presence of illumination changes and a potentially dynamic environment.

In the last few years, several approaches have been proposed in order to use background subtraction with hinged pan-tilt-zoom cameras by relying on a mosaic of the background scene. One of the heaviest drawback of background subtraction algorithms for PTZ cameras is the computational cost needed to build and maintain high quality mosaics in real time. Therefore, some approaches enact background subtraction offline [2] or propose batch surveillance applications [38]. Alternatively, real time performances have been obtained by simplifying the geometric model from projective to rigid 2D [44] or affine [41], thus limiting the fields of application to contexts in which objects are far away from the camera.

The problem of error propagation when registering sequentially a large number of images in a sequential fashion is still an open issue. Some authors dealt with it by exploiting specific informations regarding camera signals [35, 27, 29, 7], such as pan/tilt angles, or supplement the camera with additional sensors, i.e. compasses and gyroscopes.

### 3.1.2 Proposed image mosaicing method

The proposed method belongs to the class of sequential feature-based algorithms. Hence, feature detection and matching is a very critical stage, for the algorithm must be able to work fast and reliably even in cluttered and/or dynamic environments. The accuracy of the detected feature correspondences is for the overall system performance.

As regards feature detection and matching, several approaches, including the Kanade-Lucas-Tomasi tracker (KLT) [40], Harris corners [25] and the more recent Scale Invariant Feature Transform (SIFT) [31], have been tested. While SIFT demonstrated much better performance compared to KLT and Harris in terms of robustness to large interframe deformations, i.e. translation, rotation, scale and illumination changes, its computational cost greatly exceeded real-time constraints. Nonetheless, when processing a continuous video stream, differences among subsequent frames are deemed to be small.

Such consideration, along with a much lower computational cost, lead the choice over KLT and Harris methods, with KLT finally preferred because of a more stable implementation [14]. Moreover, a fast initial guess, based on a phase-correlation approach [46], is computed to assist the KLT tracker in difficult situations, namely in case of large camera shifts. The phase correlation guess serves as a coarse estimation of the camera movement and to initialize the feature tracker. Such a solution allows handling large camera displacements using small search areas, granting additional benefits in terms of robustness and performance.

**Geometric alignment**

A mosaic is a compound image built through properly composing, (*aligning*), a high number of frames and warping them into a common reference coordinate system, both spatial and tonal. The result consists of a single image of a greater resolution and spatial extent that represent a dense reconstruction of the structure and the appearance of the scene. Usually mosaicing techniques are concerned with collection of frames which do not exhibit parallax effects. Such requirement allows seamless stitching to be accomplished without requiring to recovery the underlying 3D structure of the scene. Such requirements is known to be satisfied if images are taken in either one of these two settings:

- an arbitrary scene acquired with a purely rotating camera, any rotation is allowed in place, i.e. about its optical center, no translations are allowed (to the author's knowledge this is the case of most PTZ-based surveillance applications).

- a planar scene taken from arbitrary locations.

If images are also optically corrected, i.e. as they were acquired using an ideal pin hole camera, the most general relationship between corresponding keypoints $x \leftrightarrow x^{'}$) belonging to any pair of images is described by homography matrix of Eq. 2.7.

Given a sequence of $N$ views $\{I_0, I_1, ..., I_{N-1}\}$, the construction of a mosaic requires the computation of a set of $N-1$ pairwise transformations $H_{i,j}$ that link all the views together. Assuming each image is a node in a graph and edges are homographies linking two frames, mosaicing algorithms aim at computing the homographies belonging to a spanning tree. While global registration algorithms compute all the transformations simultaneously, sequential mosaicing consists in exploring the graph one edge at a time.

Sequential algorithms usually proceeds in chronological order by determining a chain of $N-1$ pairwise homographies among images taken at subsequent instants. Hereinafter, *frame to frame* (F2F), or *pairwise* alignment, is defined as the estimation

of a homography $H_{t,t-1}$ linking a pair of temporally adjacent frames. Once the homography chain is computed, the transformation $H_{i,j}$ linking an image $I_i$ taken at time $i$ with respect to another image $I_j$ at time $j > i$ may be found by concatenating the transformations in between such as:

$$H_{i,j} = \prod_{k=i+1}^{j} H_{k,k-1} \tag{3.1}$$

By defining the reference coordinate system $R_0$ where the mosaic will be composed, $N$ visualization matrices $Q_i$ linking each image local coordinate with $R_0$ may be computed as

$$Q_i = R_0 \prod_{k=1}^{i} H_{k-1,k},\ i \in [0..N-1] \tag{3.2}$$

A mosaic can be constructed by projecting all frames $I_i$ with $i \in 0, .., N-1$ onto the common reference using the visualization matrices $Q_i$.

Sequential algorithm may also explore the graph by computing the transformations between a reference frame, usually the first, and all subsequent images. This approach is known as Frame-To-Reference (F2R). Instead of a chain, a degenerate spanning tree with one root and $N-1$ leaves describes the link topology; a set of $N-1$ pairwise homographies connecting the root with all the leaves is computed. Assuming $I_0$ to be the first frame, the transformation $H_{i,j}$ linking an image $I_i$ taken at time $i$ with respect to another image $I_j$ at time $j > i$ may be found by:

$$H_{i,j} = H_{0,i}^{-1} H_{0,j} \tag{3.3}$$

Given $R_0$, the visualization matrices $Q_i$ can be simply computed as

$$Q_i = R_0 H_{0,i} \tag{3.4}$$

Both kind of approaches have advantages and drawbacks. Frame-to-frame registration benefits from the fact that differences among temporally adjacent frames are meant to be small both in viewpoints and lighting conditions, hence keypoints correspondences are more reliable and the alignment is usually highly accurate. On the other hand since the construction of the mosaic requires all the homographies to be multiplied in a chronological order, small estimation errors propagate along the homography chain and affect all subsequent visualization matrices. As the number of frames grows, the amount of accumulated error leads to considerable misalignment. This effect is particularly noticeable when the sequence moves back and forth to the same location in the scene. When passing from the same location, frames meant to be overlapping exhibit a displacement due to the accumulated error, usually referred to as drift error.

On the other hand, Frame-To-Reference registration does not suffer from drift error since a single estimated homography is required to compute any visualization matrix.

Indeed, in case of long sequences it may happen that, at some point, a given frame do not share any overlapping areas with the reference image making it impossible to establish correspondences and compute the registration. Updating the mosaic with every new image and computing the registration between the mosaic built so far and the current image usually solves this issue; this variant is known as Frame-To-Mosaic registration (F2M). Nonetheless, long sequences still pose serious problem since fair tonal differences may arise between the mosaic and a given frame as the time pass, thwarting the keypoints matching process and, sometimes, leading the algorithm to fail and drop the frame.

Our proposed algorithm tries to get the best from both approaches by performing a dual registration stage. At first, a frame-to-frame registration between a current frame $I_t$ and the previous one $I_{t-1}$ is performed. The quality of the computed homography $H_{t,t-1}$ is then assessed according to a test involving two performance indicators:

- a normalized SSD-based similarity measure computed within the overlapping areas of the previous frame $I_{t-1}$ and the current frame warped according to the computed homography $I_t^W = H_{t,t-1}I_t$.
- the residual error $e$ of the LS estimation of homography $H_{t,t-1}$, as defined in Eq. 2.8.

If the test is passed, the computed homography is used to identify the region of the mosaic $B_t$ corresponding to the current frame and a further F2M registration step is performed between $I_t$ and the the mosaic region $B_t$. In theory, the homography $H'_{t,t-1}$ computed during the second step should be an identity matrix. In practice, it is always slightly different $H'_{t,t-1} = I_{3\times3} + \epsilon$ and its deviation $\epsilon$ helps keeping the current frame consistent with the rest of the mosaic. If the test is not passed, only the F2M registration step is performed. If it fails too, the frame is skipped.

The visualization matrix $Q_i$, relating an arbitrary frame $I_i$ to the reference frame $R_0$, is computed by alternatively multiplying F2F and F2M registration matrices:

$$Q_i = R_0 \prod_{k=1}^{i} H'_{k-1,k} H_{k-1,k},\ i \in [0..N-1] \tag{3.5}$$

The dual registration can be thought as an improved version of the Frame-to-Mosaic approach to which it return in case the first F2F registration fails. On the other hand, when F2F step succeeds the benefits from both the approaches are retained. Reliable registration with respect to the previous frame is delivered by F2F registration, cancellation of the drift error is enabled by F2M alignment. Moreover, as will be detailed in the next section, tonal registration performed after the F2F alignment bring the current frame in the photometric reference of the mosaic, thus further facilitating the F2M keypoints matching step.

Even though no theoretical analysis on the drift error reduction has been accomplished, substantial experiments have proved that our dual stage registration method is effective in bounding the amount of accumulated error and delivers quasi globally consistent mosaics. Moreover, real-time requirements are fulfilled since the algorithm is computationally equivalent to two fast sequential registration steps.

**Photometric alignment**

Tonal misalignments commonly occur when taking multiple pictures with a moving camera. If not properly handled, the resulting panorama will exhibit seams that do not correspond to any physical structure of the scene, even though the images are blended in overlapping regions. These color gradients may affect further processing involving the mosaic. For example, in a typical visual surveillance system, the motion detector based on background subtraction may erroneously interpret these artifacts as moving objects, thus generating false alarms. As a consequence, a comprehensive mosaicing technique must deal with the problem of photometric misalignments. Tonal misalignents are mostly due to:

- automatic camera exposure adjustments, i.e. changes in shutter time, auto-white balance, auto gain control and so on;
- environmental illumination changes, e.g. daytime, clouds.

Many methods have tackled the problem of exposure normalization of overlapping frames, with most of them not explicitly modeling the physical phenomena that make corresponding pixels exhibit different brightness. The works in [42, 9, 17] address the problem using spatially-varying weighting functions, also known as feathering techniques, and a clever placement of color discontinuities to minimize the visual impact, for instance along true color gradients. The seminal proposal by Burt et al. [10] on image blending using multiresolution splines have been widely employed. The idea is using a set of frequency-adaptive weighting functions by creating a band-pass pyramid representation of the image and making the transition widths a function of the pyramid level. Quite a few other methods followed on the track, anyway they tend to conceal tonal misalignments rather than correcting them and the results are visually compelling as long as the photometric difference between images is moderate.

Indeed, larger misalignments call for different and more principled approaches. The method in [11] yield remarkable results by approximating the camera nonlinear *comparametric function* with a linear piecewise function. The algorithm ultimately yield an Intensity Mapping Function (IMF) that maps every pixel brightness of a given image to the corresponding value of the tonal reference. The main drawback regards the high computational cost of such estimation, that makes it unsuited to real-time

processing. Another approach consists in the estimation of a single *high dynamic range* (HDR) radiance map built from a set of differently exposed images [26, 30, 22]. This approach models the underlying photometric process and includes an explicit treatment of saturated, both bright and dark, pixels. Once again these proposals are too time consuming and their integration into real-time systems is infeasible, at least nowadays.

Besides time performance considerations, our preferable candidate method ought to be resistant to other issues, for example spatial registration inaccuracies, arising from small alignment errors, and the presence of moving objects in the scene. These further considerations prompted us to exploit an histogram-based approach, that allows to partly overcome the above mentioned problems. The histogram specification (HS) technique is a histogram-based approach that aims at transforming a cumulative distribution $H_1$ of a random variable into the cumulative distribution $H_2$ of another random variable by finding a continuous remapping function (see [21] for further details). Assuming a given image and its tonal reference as two random variables, the remapping of the brightness value of each pixels of the image according to the computed function results in the given image histogram matching the tonal reference one. In this context, the remapping function is named *Intensity Mapping function* . If the image at hand and its reference are properly spatially aligned, identical histograms yields photometric alignment. Unfortunately, exact histogram specification holds only for continuous random variables whereas pixel brightness is not. Nonetheless many algorithms, such as [13, 23], have been conceived to approach theoretical performances. A more in-depth presentation of related concepts and topics is postponed to the next section 3.2.

Anyway, a typical IMF for gray scale images is a discrete function consisting of 256 pair of corresponding pixel brightness ($u_1$, $u_2$) derived from the cumulative histogram $H_1$ and $H_2$ of a given image and its tonal reference as follows:

$$u_2 = H_2^{-1}(H_1(u_1)) \tag{3.6}$$

A specific photometric registration method relying on the histogram specification technique is part of the proposed mosaicing approach. This color normalization step is performed prior to stitch a new frame into the mosaic, just as the geometric registration step aligns the images into a common spatial coordinate frame. Based on HS, the method is fast and simple; moreover it does not require the scene to remain completely static and is tolerant against moving objects and small spatial registration errors. In fact, the presence of few moving objects often does not alter the overall cumulative histograms hence impacting negligibly on the photometric registration stage.

Although the method has been conceived to work with gray scale images, its practical generalization to color imagery has been accomplished by transforming images into a luminance-chrominance color space, such as YUV space, then perform histogram

specification on the intensity channel, apply the IMF and transform back. Performing histogram specification independently on each channel of a RGB color image may cause tonal artifacts such as the introduction of color hues absent from both source and target color schemes, as might be seen in Fig. 3.1. This effect probably originates from the use of Bayer color filter array, which is a popular format for digital acquisition of color images, and might have a smaller effect when using full color CCD camera (3 independent photo receptors per pixel, one for each RGB channel). More details on the histogram specification topic and a fast implementation of the algorithm are given inside [4].

A principled extension of histogram specification to color images has been attempted in [33, 32], conversely a biologically inspired approach that handles the correlation between color channels and their perceptually non uniformity is still to come.

**Figure 3.1:** Independent histogram specification on each channel of the RGB color space. Unexpected hues appear due to inaccurate correction of photometric misalignment.

### Image warping and blending

Every new frame $I_i$ is combined into the mosaic by warping it according to the visualization matrix $Q_i$ computed by the geometric alignment stage. Image warping is accomplished using the backward transformation, namely for each destination pixel its corresponding source pixels color is queried. In this way neither holes nor overlaps

can appear in the warped image, and inside the mosaic accordingly. The backward approach requires the inverse of the visualization matrix $Q_i$ to be computed, anyway the matrix $Q_i^{-1}$ always exists since homographies are non singular linear transformations.

Several different interpolation methods have been investigated among those suggested in literature. In the experiments, bilinear interpolation has been chosen as it has empirically proved to offer the best tradeoff between accuracy and computational cost with respect to higher order methods (e.g. cubic interpolation). Conversely, nearest neighbor interpolation exhibits too much visual artifacts, otherwise it would be attractive due to its speed. Photometric registration, accomplished through a simple pixelwise Look-Up Table (LUT) recoloring using the computed IMF, is performed prior image warping.

Although geometric and photometric registration should take each frame into the spatial and tonal reference of the mosaic, seamless stitching usually calls for an additional blending stage in order to conceal small residual artifacts. Blending techniques consist of a filtering process inside the overlapping areas, usually attained by means of weighting functions that reinforce smoothness or continuity among adjacent pixels or regions [39, 10]. Different approaches may encompass temporal filtering schemes such as mean, mode or median of the distribution of overlapping pixels [6], or also the exponential update rule. Statistical approaches model the color distribution at each pixel using parametric [28] or non parametric mixture of gaussian [18].

Since the proposed algorithm usually leave faint residual artifacts, a simple and fast blending method based on modal filtering has been preferred. In practice, the mode of the intensity distribution of each pixel is considered the representative sample and selected to appear into the mosaic image. Assuming pixel intensities being affected by gaussian noise, this approach is close to a maximum likelihood estimation, anyway it empirically proved to be robust with respect to small misalignments and undetected foreground objects.

In presence of moving objects detected by the background subtraction algorithm (more on that in the following section), a selective update is enabled in order to use the computed masks as filters to prevent the update of parts of mosaic currently occluded by foreground objects. Obviously, when a new frame observes unseen areas of the mosaic, no previous information to perform background subtraction is available, hence pixels belonging to new areas are assigned to the mosaic directly.

### 3.1.3 Motion detection

A reliable background mosaic permits to directly extend the use of a standard background subtraction algorithm for stationary cameras, for example the work presented in [6], to moving PTZ cameras. Although the explanation of the concepts underpin-

ning background subtraction algorithms is outside of the scope of this section, standard methods basically compare the current frame $I_i$ with a reference image $B$, i.e. a previously computed background. Moving "blobs", or aggregate of pixels, are identified by thresholding the result of the comparison. A moving PTZ camera does not admit the equivalent of a reference image for it is allowed to change its viewpoint over time. While recording all possible portions of observable scene may not be practical, combining all the views in a single representation can be accomplished by constructing a mosaic of the scene. Indeed, standard background subtraction algorithms can still be employed as long as a prior step trim the portion of the currently visible scene, the background $B$, from the mosaic and feed it to the motion segmentation algorithm.

Although, in principle the mosaicing algorithm may be used once for the creation of the background mosaic during a bootstrap sequence and then left unused, this is not recommended. As a matter of fact, for the background subtraction and maintenance operations to be performed efficiently, the current visualization matrix $Q_i$, linking the current image to the mosaic should always be kept up-to-date. For this reason registration is performed at every new frame even though the background mosaic is already in place. This way, the current visualization matrix $Q_i$ holds the position of the frame inside the mosaic image, or, equivalently, the location of the corresponding region $B_i$.

After the portion of the currently visible background $B_t$ has been indexed, the alignment with the actual frame $I_i$ is easily accomplished by backprojecting $B_i$ using the inverse of the current visualization matrix $Q_{i-1}$.

As a final remark, the exploitation of color images permits to achieve considerable improvements in terms of shadow removal and reduction of camouflage, i.e. whereas different color tuples map to similar gray level values, although requiring an increased demand of computational resources. In particular, performing background subtraction in a different color space, such as HSV or $YC_rC_b$, permits to reveal moving shadows and to discard them when detecting motion [15, 16]. Shadows can have very a detrimental effect, especially in outdoor environments, causing deformations of the shapes of moving objects that lead to degraded results of further processing tasks such as tracking or object recognition.

### 3.1.4 Experiments

Extensive experiments using several video sequences captured from real world scenes have been accomplished in order to evaluate the quality of the mosaics generated by the proposed algorithm. Since no standard evaluation methodology nor sequence dataset are available, quality assessment is mostly delegated to visual inspection. Nonetheless the integration within a visual surveillance system allows to consider the overall system performance, namely the computed motion masks, as an indicator of the mosaics

quality as well.

To this purpose this section is subdivided into two parts. The first part focuses on the visually perceived quality of the mosaics; despite being a subjective indicator it provides substantial insights as long as inherently "qualitative" applications are targeted, such as digital photography, photomontage, post production effects and so on. The second part is concerned with motion segmentation using a PTZ surveillance camera and aims at assessing the performance of the algorithm by examining the quality of the motion masks delivered by the overall system.

A considerable number of image sequences have been used throughout this section, all of them being different for many specific aspects such as length, environment, illumination, moving objects and so on. Though, the resolution, $320 \times 240$ pixels, and the processing hardware, an AMD 2000 MHz, is the same for all of them. For this reason, time performance delivered by the mosaicing algorithm are quite stable, irrespectively of the specific sequence, and fluctuates in the range of $10 - 15$ frames per second (FPS) for gray scale images and $5 - 9$ frames per second for color RGB images. Such processing speed allows the motion detection system to perform adequately smooth and to deliver the expected people tracking and alarm signaling functionalities.

**Image mosaicing results**

Four image sequences have been selected to illustrate the visual quality of the attained mosaics. All the sequences consists of several hundreds of frames and have been acquired by moving a camera around without particular care. The first pair of sequences require spatial alignment only, the second pair tonal alignment as well. In Figure 3.2 two mosaics, attained by processing the first pair of outdoor (top) and indoor (bottom) sequences, are shown.

The first outdoor sequence DCOURT1 (Figure 3.2 top) consists of 680 stills and has been acquired by manually scanning the scene from left to right and back many times. The scene exhibit objects at a variable distance from the camera; e.g. a close wall of a building on the left, a farther gate and a paved courtyard. The wide field of view and the structured scene (hedges and trees) may emphasize small alignment errors. Nonetheless the mosaic does not exhibit any visible artifacts or seams, all the structures being properly aligned and uniformly colored.

The second sequence DLAB1 (Figure 3.2 bottom) is 820 frames long and portrays an indoor environment with very close objects. The small distance between the observer and the surroundings makes the assumption of quasi-flat scene hardly fulfilled, leading parallax effects to hinder motion parameter estimation. Nonetheless, the mosaic does not contain any blur or discontinuity and the texture of the scene is sharp and in-focus everywhere.

**Figure 3.2:** Two examples of mosaic built from long sequences acquired by randomly panning the camera back and forth across the scene. (Top) Mosaic from sequence DCOURT1 (680 frames). (Bottom) Mosaic from sequence DLAB (820 frames).

The second pair of sequences is more challenging. Aside the more complex camera motion trajectory, as may be realized by the irregular shapes of the attained mosaic, considerable illumination changes have taken place during the acquisition. In fact, the scope is to highlight the visual quality improvements the proposed mosaic delivers by explicitly compensating illumination changes.

The first sequence, DCOURT2, has been acquired at 12.5 fps with a remote controlled Axis PTZ network camera pointing toward the same outdoor scene as is sequence DCOURT1, but taken from a different point of view. Several exposure changes occur along the sequence due to the automatic light compensation mechanism embedded in the camera firmware. An example of a sudden photometric variation may be appraised by looking at Fig. 3.3 where in a matter of few frames the image becomes highly saturated.

As shown in Fig. 3.4 (top), although the proposed spatial registration algorithm manages to preserve the consistence of the geometric structures across the whole scene, many visually unpleasant seams show up due to the considerable tonal misalignment among frames. Conversely, all the artifacts are eliminated by enabling the photometric

**Figure 3.3:** Department courtyard (DCOURT2) sequence: pair of temporally adjacent frames with strong photometric variations.

registration and a smooth and sharp reconstruction is obtained, as can be seen in in Fig. 3.4 (bottom).

The last sequence, DLAB1, deals with an indoor highly structured environment. As before, the sequence has been acquired by manually panning and tilting, using a Sony TRV 900 camcorder hinged on a tripod. Spot lights spread across the scene cause sudden exposure compensation every time the camera directly points at them. The effect of uncorrected photometric changes, shown in Fig. 3.5 (top) seriously degrade the quality of the mosaic. However, when tonal registration is performed, most of the color defects disappear and the outcoming mosaic looks much more pleasant and realistic (Fig. 3.5, bottom).

**Motion detection results**

Indirect assessment of mosaic quality through the analysis of the performance delivered by a visual surveillance system is the scope of this section. The motion masks computed by the motion segmentation algorithm have been visually inspected, Receiving Operator Characteristic (ROC) or other statistical indicators being impractical since no public data sets equipped with ground truth, for these kind of applications, are available yet.

All the sequences have been captured with a Sony TRV 900 camcorder at about 12 frame per second (fps) and $320 \times 240$ pixel resolution. The camcorder has been hinged on a tripod in order to make it rotate roughly about its optical center. Six challenging indoor and outdoor sequences have been considered, being different for illumination, scene structure, and number of moving objects in the scene.

The first sequence DLAB2 is 1121 frames long; it is the sequel of sequence DLAB1 and consists of a wide field of view capture of the interior of our lab. In Fig.3.6, both the mosaic (top) and the plan of the environment (bottom) are reported. Similarly to sequence DLAB1, close objects and significant depth variations (near the red door

**Figure 3.4:** Department courtyard (DCOURT2) sequence: spatially aligned mosaic (top), spatially and tonally aligned mosaic (bottom).

and the wall on both sides) may emphasize slight out-of-center rotations giving raise to disturbing parallax effects. Moreover the vicinity of the moving foreground object requires fast camera rotation to allow person tracking, hence leading to large interframe shift typically difficult to handle.

Despite the mentioned difficulties, the system performed consistently and accurately. Samples of the delivered motion masks are superimposed on the frames in Fig.

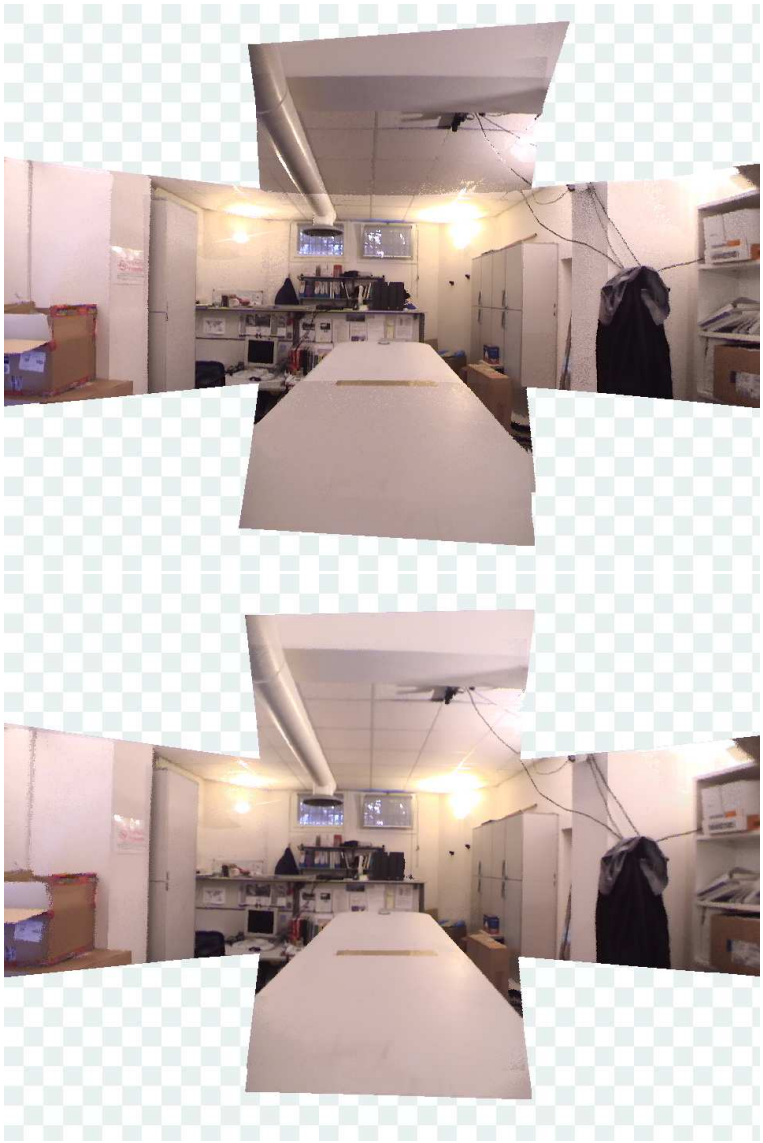


**Figure 3.5:** Department laboratory (DLAB1) sequence: spatially aligned mosaic (top), spatially and tonally aligned mosaic (bottom).

3.7 to ease visual inspection. Detected moving blobs are adherent to the real body shape of the moving person across the entire sequence irrespective of its position inside the scene and its distance from the camera.

The second sequence DCOURT3 deals with an outdoor environment with a person walking in (see Fig. 3.8 and Fig. 3.9). The scene structure is favorable since the wall is perfectly flat, on the other hand reliable feature detection and matching is difficult

**Figure 3.6:** Mosaic created through processing the indoor sequence DLAB2 (top), plan of the environment and cone of view(bottom).

for the building being poorly textured. Moreover, the proximity of the moving person cause large interframe displacements stressing further the KLT tracker.

Nonetheless, the initial estimation via phase correlation effectively supplements the KLT tracker leading to reliable estimation of the transformation parameters. As a result, the detected moving masks reflect the presence of the person and provide a good approximation of the real shape, as it might be seen in Fig. 3.9. Few false detections appear from time to time, due to the shadow cast on the wall behind.

A third and more challenging sequence, DCOURT4, consists of 1457 frames and deals with the large outdoor environment partly visible in Fig. 3.4. Three walking

**Figure 3.7:** Motion detection sample frames from the indoor sequence DLAB2.

person roam around randomly (see Fig. 3.11) and the camera follow their movements. Several difficulties arise when processing such a sequence. Firstly, although foreground objects moves slowly because of the distance, LAN traffic caused frame drops and hence the sequence shows a highly variable frame rate including significant frame lag. Secondly, the scene exhibit a large range of radiance, the courtyard on the lower right side being far more darker than the sunlit buildings within the upper area. As the camera rotates through the scene, such highly varying illumination conditions need to be compensated to avoid seams in the resulting mosaic. Finally, three moving person are simultaneously present in the scene. By moving independently from the camera motion, a large number of keypoints detected on the objects becomes motion outliers possibly degrading RANSAC performance.

Nonetheless, the motion detector performs steadily, regardless of the lighting condition and the distance of the moving objects. The achieved moving masks adhere to the silhouette of moving objects, although often signaling also their cast shadows (see Fig. 3.11).

The next two sequences focus on the benefits deriving from color processing. The most remarkable advantage, as long as motion segmentation applications are concerned, is the ability to remove shadows using intensity-cromaticity color spaces. In fact, a color mosaic can trigger the use of different color spaces to get a significant improvement of the motion detection outcomes. An example of the improvement granted by performing background subtraction in the $HUV$ space compared to gray scale is

**Figure 3.8:** Mosaic created through processing the outdoor sequence DCOURT3 (top), plan of the environment and cone of view(bottom).

shown in Figures 3.12 and 3.13.

In rows of Figures 3.12 and 3.13 one can see three frames extracted by two sequences showing the output of the motion detection referring to the same environments (DLAB1 and DCOURT1, respectively) depicted in Figure 3.2. As always, the detected moving masks have been superimposed to the frames to ease the visual inspection. In the top row, the quality of the detected masks using conventional gray scale frames is presented. In the bottom row, it is shown the improvement yielded by exploiting color information.

In the samples depicted in Figure 3.12, a person enters the room and casts his shadow on the wall behind (left), conversely the shadow is removed when using chromaticity (right). In the second set of samples depicted in Figure 3.13 a walking person is moving around in a sunlit courtyard. Being an outdoor scene, the shadow is yet more highlighted compared to the indoor one. Although it is clearly visible in the gray level

**Figure 3.9:** Motion detection sample frames from the outdoor sequence DCOURT3.

**Figure 3.10:** Mosaic created through processing the outdoor sequence DCOURT4.

sequence (left), it has been completely removed in the color one (right).

The last sequence shows the impact of accurate tonal alignment on both background subtraction and tracking performance of the visual surveillance system. Fig. 3.14 shows a couple of frames referring to the same environments depicted in Fig. 3.4 and highlights the motion masks; objects identities, computed by the tracking algorithm,

**Figure 3.11:** Motion detection sample frames from the outdoor sequence DCOURT4.

**Figure 3.12:** Three gray scale (top) and color (bottom) sequences of three frames each, coming from DLAB sequence showing shadow suppression using color imagery.

are visualized by means of different colors. Moreover, motion segmentation information are superimposed along with the trajectory followed by the moving object during the last 20 frames.

Due to unhandled illumination changes, sample frames on the left column of Fig. 3.14 depicts highly inaccurate motion masks yielding to perturbed motion trajectory. On the contrary, photometric correction allows to deliver reliable motion masks and accurate trajectories accordingly. As an example Fig. 3.14, middle left, shows a large artifact

**Figure 3.13:** Three gray scale (top) and color (bottom) sequences of three frames each, coming from DCOURT sequence showing shadow suppression using color imagery.

in the middle of the image, which yield the system to detect one insgle moving mask instead of two. In this case, a potential detection error is fixed by the tracking algorithm that recognize the two persons despite the single detected mask. Besides, on the top right side of the left image a big false alarm is triggered. Conversely, on the middle-right, motion masks are detected with a quality comparable to that of background subtraction with stationary camera. Such a quality enables reliably objects tracking in the whole field of view, independently of the camera movements.

### 3.1.5 Summary and future work

An automatic, real time and general purpose image mosaicing algorithm has been conceived. The proposed method performs consistently in a wide range of real world contexts, e.g. indoor and outdoor scenes, by deploying an explicit spatial and tonal registration procedure. In addition the system is completely image-based and it does not rely on any a priori assumption regarding scene or camera.

The dual alignment stage permits to bound the drift error allowing the construction of quasi globally consistent mosaics, without resorting to computational demanding global adjustment procedures. The use of fast features, supplemented by a phase correlation based bootstrap, permits to handle large and complex camera motions while preserving real-time computation. The accuracy and the high processing speed make the algorithm suitable for integration in visual surveillance systems performing on-line motion detection using background difference. Experiments with several challenging real-world video sequences have shown the effectiveness of the proposed approach for both visual and quantitative purposes.

As for future works, the system may be improved by adding on-line learning of

**Figure 3.14:** Department Courtyard (DCOURT2) motion detection and tracking sample frames: with (right) and without (left) joint spatial and tonal alignment.

optical properties (focal length, principal point and lens distortions) of the imaging device. The correction of optical non-idealities would lead to a complete independence from the imaging device and would considerably enhance both spatial and tonal alignment. In addition, a faster implementation of SIFT features will provide more reliable feature correspondences, and a more accurate stitching accordingly.

## 3.2 A fast and exact histogram specification method

Histogram specification methods aims at finding a function that transforms a source image so as to match a target distribution with the highest possible degree of accuracy. Many approaches privilege exact specification by exploiting multi-valued ordering functions but incur in computationally expensive implementations. Aside computational complexity, histogram specification algorithms can be rated according to image distortion and accuracy of reproduction of the target histogram, i.e. histogram matching.

Topic of this section is a fast algorithm, based on histogram specification, that deliver exact matching to a given target histogram independently of the source image meanwhile introducing negligible image distortion. The simplicity of the method enables fast computation making the algorithm suitable for real time applications, such as sequential image mosaicing.

### 3.2.1 Introduction

Histogram modeling techniques provide sophisticated methods for manipulating colors and contrast of an image by altering individual pixel such that the intensity histogram assumes a desired shape ([34, 20]). Histogram specification is a basic histogram modeling technique that transform one histogram into another one by remapping pixel brightness values according to a computed Intensity Mapping Function (IMF). Although histogram modeling operators may encompass the use of complex IMF, histogram specification employs a simple monotonic, non-parametric mapping which re-assigns the intensity values of pixels in the input image such that the output image exhibits as a similar histogram as possible to a given target distribution. Ideally, target and output image histograms should be as similar as possible.

Although in a theoretical continuous case a mapping function yielding a desired Probability Distribution Function (PDF) exists, in the discrete domain of pixel brightness values only approximated IMF can usually be determined. Approximated IMFs produce quasi exact histogram matching by introducing well known histogram artifacts such as gaps and overfull bins, but preserve image structures.

Classic algorithms ([34, 20]), relying on approximted IMFs, have been used for a wide range of tasks where visual evaluation is crucial, due to them preserving as much the image structures as possible. On the other hand, histogram artifacts can have very detrimental effects for subsequent image processing operations such as image fusion, invisible watermarking, image normalization and image mosaicing.

Recent researches in the field of histogram specification has led to diverse approaches aiming at lower histogram distortions, by slightly modifying the image struc-

ture. Quasi exact and exact specification has been achieved by exploiting multi-valued IMF capable of mapping pixels according to diverse features, i.e. pixel brightness, average neighborhood brightness, thus allowing to diminish histogram distortions. Though, the determination and the mapping using multi valued IMF require computationally expensive algorithms.

This section presents a novel approach for fast and exact histogram specification. The conceived method delivers exact histogram matching meanwhile introducing low image distortion and allowing for fast computation. The main novelty is the use of one-to-many (OTM) relations among source and target pixels brightness instead of standard one-to-one mapping. This yields a quick and flexible remapping policy able to prevent any histogram distortion.

### 3.2.2 Related work

Histogram specification ([34, 20]) may be regarded as a generalization of histogram equalization ([34, 37]). Classic implementations of histogram equalization rely on the fact that transforming a Random Variable (RV) by its Cumulative Distribution Function (CDF) results in a uniform distribution ([20]). Histogram specification is performed by using the source CDF to map the source histogram to a uniform one and then using the inverse of the target CDF to make the uniform histogram to reproduce the target one.

By modeling pixel brightness as a discrete RV $r$ characterized by a PDF $p_r$ that describes the spatial frequency of its gray levels, it can be shown that the RV $R = C_r(r)$ is *uniformly* distributed in $[0, 1]$, where $C_r(r) = \int_{-\infty}^{r} p_r(v)dv$ is the monotonically non-decreasing CDF of $r$. Besides, let $Z = C_z(z)$, $z$ and $Z$ being RVs and $C_z$ the CDF of $z$, then one can force $R = Z$, hence $z = C_z^{-1}(Z) = C_z^{-1}(C_r(r))$, as long as $R$ and $Z$ are uniform. Then, it turns out that $C_r$ and $C_z^{-1}$ are the equalizing and the reshaping function, respectively. Apart from normalization details, histogram specification is performed by replacing source image graylevel $r$ with $r'$: $r \rightarrow r' = C_z^{-1}(C_r(r))$.

While, in the continuous case, a function capable of transferring the PDF of a target image to a source image exists, in a discrete domain the same problem usually admits only approximated solutions. This is due to discrete CDFs being not exactly invertible, for they are staircase functions and therefore invertible when pixels take distinct values only. Since the number of pixels in an image is usually considerably larger than the number of graylevels, the distinct value case is unlikely to occur.

Classic specification algorithm [34, 20], implemented according to the above described theoretical framework, discriminate pixels according to their brightness value, thus leading to quasi exact histogram matching, with the delivered histogram affected by artifacts such as holes and overfull bins. Despite producing histogram distortion, these algorithms are fast and introduce low distortion of image structures, for pixels

showing the same graylevel in the original image being mapped into the same target graylevel. Fig. 3.15 shows an example of the distortion affecting a histogram delivered

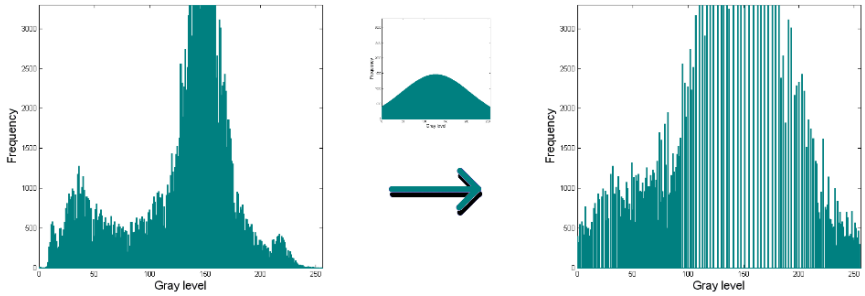



**Figure 3.15:** An image histogram (left), a target histogram (middle) and the outcome after classical specification (right)

by classic histogram specification algorithms.

Histogram distortion artifacts, i.e. gaps and overfull bins, originates when the derivatives of corresponding ranks of source and target CDFs exhibit different values. In this cases both matching ambiguities and overassignments could arise. In practical cases, gaps and overfull bins are emphasized in case the source image histogram is composed by few large bins. For these reason, the authors of [34, 37] propose to reduce this effect by preprocessing the source image adding a small amount of uniform noise, so as to avoid large bins. While these approaches are likely to produce an output image whose histogram is more similar to the target one, the randomly added noise may potentially reduce the overall image quality by degrading image structures. Nonetheless, these methods grant an improvement in terms of histogram matching compared to classic methods, and the unstructured noise may be filtered by further processing procedures.

Several other attempts have been accomplished to improve histogram matching by exploiting methodological techniques. For instance, the authors of [43] reformulate the histogram specification problem as an optimization problem. However, exact matching is still attained at the expense of noisy images, as noted in [34]. Moreover, this method introduces structured noise patterns, i.e. horizontal lines inside uniform areas, due to the row-wise order of evaluation of equivalent pixels. As a matter of fact, such patterns, although sometimes visually negligible, may mislead further image processing methods (e.g. edge detectors), whereas noisy lines might be mistaken for real scene structure. The use of multi-valued IMF has been pioneered by th work of Hall [24], where the histogram approximation has been improved by further discriminating

pixels according to the local average of the 4-connected neighborhood. Recently, this work has been refined by other authors. For instance, Eramian et al. [19] proposed two novel neighborhood based metrics to separate pixels with same graylevel, e.g. the 8-connected average and the brigther-than-neighbors count. While this approach permits to effectively split larger bins into smaller ones, exact histogram matching is not always secured. Coltuc et al. [13] further improve the latter approaches by combining different metrics, using a variable length bank filtering approach, with the purpose of discriminating *each* pixel of the image. Uniquely indexing every single frame amounts at obtainig invertible CDFs, thus making the exact solution to exist, as it happens in the continuous case. Leaving the computational complexity of the method apart, the choice of the filters plays a key role in the indexing process, The ability to discriminate every single pixels can be attained by analyzing image properties inside large windows centered on each pixels. On the other hand, features extracted from regions far away from the given pixel may provide loosely correlated information. Often, the right filter size is strictly dependent on image peculiarities and it must be carefully chosen to prevent the computational cost to diverge. Nonetheless, the work reported in [13] represents the state-of-the-art for exact histogram specification methods.

### 3.2.3 The method

Histogram specification methods can be classified according to computational complexity, image distortion and accuracy in reproducing the target histogram. The proposed method yields histograms perfectly matching a target PDF meanwhile introducing low image distortions.

#### Approaches

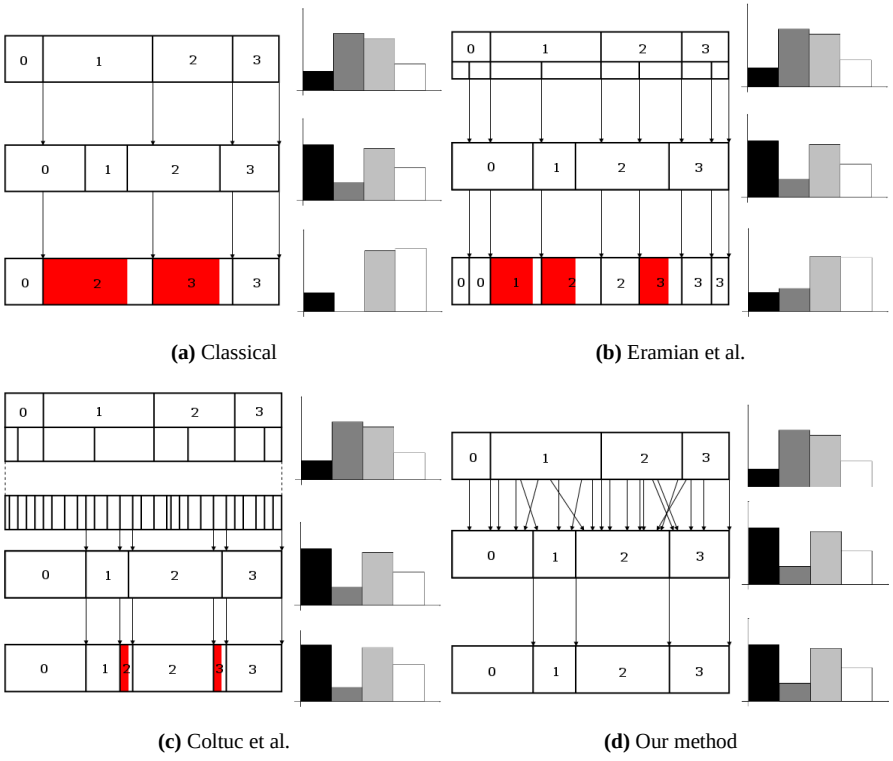
According to theory in Section 3.2.2, histogram specification is generally accomplished through a mapping between order statistics, where each element of the source distribution is mapped to the correspondingly ranked element of the target distribution. Thus let $f : [0, N-1] \times [0, M-1] \rightarrow [0, D-1]$ be a scalar function representing an image with dimension $E = N \times M$ and depth $D$, where $f(p)$ denotes the graylevel of a pixel $p$. In this setting, the discrete PDF $H_f$ (i.e. the normalized histogram) and the CDF $C_f$ of the image $f(\cdot)$ can be computed as follows:

$$H_f(x) = \frac{1}{E} \sum_{p=0}^{E} S(f(p)), S(f(p)) = \begin{cases} 1 \text{ , if } f(p) = x \\ 0 \text{ , otherwise} \end{cases} \tag{3.7}$$

$$C_f(x) = \sum_{y=0}^{x} H_f(y) \tag{3.8}$$

The ideal output of a histogram specification algorithm is an image $g : [0, N-1] \times [0, M-1] \rightarrow [0, D-1]$ with a normalized histogram $H_g$ that exactly matches the target PDF $H_t$. Given these definitions, each bin of the output histogram must count $N \times M \times H_t(i)$ pixels, where $i \in [0..D-1]$ represents the bin index.

Fig. 3.16 outlines graphically, using only 4 gray levels, the way gray levels are remapped to perform histogram specification by the algorithms described in [20, 19, 13] and our proposal. The first rows refer to the source image and show the distribution of the 4 gray levels (left) with the corresponding histogram (right). The second rows show the target distribution (and related histogram), while the arrows from first to second rows describe the re-mapping procedure (e.g. in Fig. 3.16(a) 0 maps to 0, 1 to 2, 2 to 3 and 3 to 3). Finally, the third rows show in red (dark) color the approximation errors. For example, the third row of Fig. 3.16(a) shows that gray level 1 is mapped erroneously in gray level 2 instead that partly in 0, 1 and 2.



**(a)** Classical **(b)** Eramian et al.

**(c)** Coltuc et al. **(d)** Our method

**Figure 3.16:** Histogram specification mappings methods and approximation errors. Classical (a), Eramian et al. (b), Coltuc et al. (c), our method (d)

Fig. 3.16(a) depicts the classic histogram specification algorithm. It requires to compute a simple IMF in the form of a Look-Up-Table (LUT) whose entries refer to the $D$ distinct pairs $x_i \leftrightarrow x_i^{'}$ where $x_i$ is a source gray level and $x_i^{'}$ is a target gray level.

Hence, $D$ represents the dynamic range of the implicit ordering function based on gray levels only. In order to perform histogram specification each graylevel $x$ is replaced with the target graylevel $x'$ according to:

$$x' = D \cdot C_T^{-1}(C_S\left(\frac{x}{D}\right)) \tag{3.9}$$

where $C_t$ and $C_s$ are the CDF of the target and the source histogram, respectively. This method is simply a graylevel remapping, only global histogram information and the pixel graylevel are considered. The more source and target histograms are different, the more gaps and overfull bins are likely to appear. In fact, large difference in pixels count of corresponding bins, through the computed mapping, may cause assignment problems. The issue is originated from the staircase nature of the discrete CDF and the coarse quantization step $\epsilon = 1/D$ given by discriminating pixels only on the basis of brightness values. As highlighted in Fig. 3.16(a), this problem may lead to gross approximation errors and poorly matching histograms.

An attractive improvement arises from discriminating pixels having the same gray level, taking into account some properties of image neighborhood. For example, authors in [19] introduce the neighborhood voting metric $\alpha$, defined as a function of the number of pixels in the $m \times m$ square neighborhood mask centered on a pixel whose gray value is strictly less than the pixel brightness. Formerly equivalent pixels can be further distinguished in $m \times m$ classes according to the metric. Thus the dynamic range of the ordering function based on brightness and metric $m_\alpha$ amounts at $D_\alpha = D \cdot (m \times m)$. This grants a finer quantization step $\epsilon_\alpha = 1/D_\alpha$. In practice, equal gray level pixels may be discriminated into additional $m \times m$ bins, thus reducing the staircase effect of the CDF and yielding a better approximation of the desired histogram (Fig. 3.16(b)).

Along the same line, another proposal by the same authors of [19] concerns an algorithm relying on a metric $\beta$ defined as the $m \times m$ neighbor average brightness around each pixel, approximated to the nearest integer. Similarly, equivalent pixels can potentially be further subdivided in $D$ classes, thus resulting in a dynamic range of $D_\beta = D \cdot D$.

In principle, several metric, or features, may be added until each bin consists, at most, of a single pixel. An interesting example is the work by Coltuc [13] that combines $K$ average neighborhood metric $m_k, k \in [0..K-1]$ computed on image neighbors of increasing size, thus yielding a dynamic range $D_M = \prod_{k=0}^{K-1} D_{m_k}$. As one can see in Fig. 3.16(c), the quantization step decreases and the histogram converges significantly to the target one.

**Description of the algorithm**

The proposed algorithm has been primarily designed to meet the definition of histogram specification, namely the generation of an image whose histogram perfectly matches a given target histogram, independently of the source image. Since each bin $i$ in the output image must be populated with exactly $E \times H_T(i)$ pixels, it is likely to happen that pixels having the same source graylevel shall be spread to different target gray levels. Nonetheless, the case of indistinguishable pixels may occur irrespective of the dynamic range of the conceived ordering function.

Therefore, standard IMF, namely bijective relation, has been abandoned in favor of the concept of one-to-many *relationship*. A one-to-many relationship holds the potential to handle indistinguishable source pixels by explicitly modeling their mapping to multiple target graylevels. In place of fixed one-to-one correspondences $x_i \leftrightarrow x_i^{'}$, given by a conventional IMF, one-to-many relations allow to assign a given source pixel many target values inside an admissible range $x_i \leftrightarrow \left(x_i^{'}, x_i^{''}, ...\right)$. Final gray level assignment is drawn randomly inside every admissible range, although ensuring exact histogram matching. Moreover, the proposed method deliver exact specification with any ordering function; the use of other metrics, in addition to the brightness value, affect only the size of the admissible ranges.

Pixels are first ordered according to a given ordering metric, yielding several classes of equivalence $c_i$, $i \in [0..D_M-1]$, e.g. defined by individual brightness values. Nonetheless, as mentioned in the previous section, a class of pixels can be further split into subclasses according to other properties, e.g. neighbor brightness average. More different properties yield more subclasses, thus producing a finer quantization step.

After equivalence classes have been computed, each subclass $c_i$ is sequentially assigned to target gray level bins $b_j$, $j \in [0..D-1]$ so that each of them have $E \times H_T(i)$ items. A sparse matrix $M_{D_M \times D}$ stores the one-to-many mapping, in which row represent source image class and the columns denote target histogram graylevels. Each matrix entry $M(c_i, b_j)$ contains the amount of source pixels inside a class $c_i$ that must be remapped to the gray level $b_j$:

$$\begin{bmatrix} m_{0,0} & 0 & 0 & 0 & \dots & 0 \\ m_{1,0} & m_{1,1} & 0 & 0 & \dots & 0 \\ 0 & m_{2,1} & m_{2,2} & 0 & \dots & 0 \\ 0 & 0 & m_{3,2} & 0 & \dots & 0 \\ 0 & 0 & m_{4,2} & m_{4,3} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \ddots & m_{D_M,D} \end{bmatrix} \tag{3.10}$$

Histogram matching is attained by imposing that each target gray level (column entry)

is assigned $E \times H_T(i)$ source pixels or, equivalently, by requiring each column to sum up to $E \times H_T(i)$ pixels

$$S(b_j) = \sum_{c_i=0}^{D_M-1} M(c_i, b_j) = E \times H_T(j), \forall j \in [0..D-1] \tag{3.11}$$

As many other approaches, this method may introduce structured pattern noise in case gray level assignments follow the order of the evaluation of the input pixels (e.g. typically row-wise). However, although the visual effect can be noticed only for quite untextured and largely uniform images, this always alters the signals in a systematic way, possibly misleading further image processing algorithms. This effect has been significantly alleviated with the introduction of random strings for shuffling gray level $b_j$ inside a given admissible range before establishing the final assignment. Being computed offline, random strings avoid to add misleading signal patterns to the output image while preserving computational efficiency.

### 3.2.4 Experimental results

Extensive experiments have been carried out using standard images widely employed for benchmark evaluations. In addition, challenging images have been considered in order to stress the considered methods and emphasize the outcome of the different strategies adopted. The target machine is a AMD Athlon 2000+ equipped with 512 MB RAM.

Three quantitative performance indicators have been considered, thus allowing even small differences to be highlighted. In particular, comparisons have been performed according to computing speed and contrast enhancement. However, as stated at the beginning, a poor histogram matching can affect further image processing steps even when it is not perceivable. To this purpose, two distortion indicators measure to which extent the histogram and the image structure have been altered by the specification process.

The experiments have been accomplished over the most four representative methods in literature and results have been compared with the outcome of the proposed approach. The names of authors in the Tables refer to the methods described in the respective papers. In particular, Coltuc [13] is the only method achieving an exact matching histogram, and it is the most direct competitor for all indicators. Finally, for one image we include the shapes of the original and specified histograms, using all the methods implemented.

**Performance Indicators**

In this section the performance indicators used to assess and compare the histogram specification methods are detailed:

- *computational speed*
  Often the time needed to obtain the specified histogram is not directly measurable since the elapsed time is too short. Therefore we have computed the number of specifications performed in a given amount of time, that in the experiments has been fixed to 10 seconds. In this way we can derive the number of iterations per second $S$ that is our figure of merit.

- *histogram distortion*
  This indicator gives a measure of the effectiveness to achieve a specified histogram by comparing output and target histograms, $H$ and $K$ respectively, by using the Kolmogorov-Smirnov distance defined in Eq. 3.12:

$$D_{KS}(H,K) = \max_i(|\hat{h}_i - \hat{k}_i|), i \in [0..D] \tag{3.12}$$

  where $\hat{h}_i$ and $\hat{k}_i$ represents the $i^{th}$ bin of the histograms.

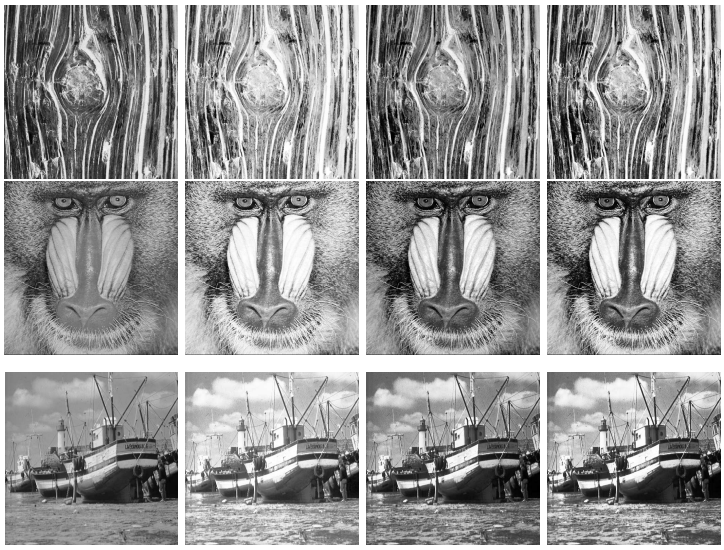
- *image distortion*
  Among the possible indicators to measure image distortion we have chosen the one implemented by authors in [19], in order to better allow a direct comparison. The image distortion between images $G_1$ and $G_2$, whose size is $N \times M$ has been measured according to the following indicator.

$$\Delta = \frac{1}{E} \sum_{(i,j)\in[0,M-1]\times[0,N-1]} \left( \frac{G_1(i,j)}{G_2(i,j)} - \mu_{ij} \right)^2 \tag{3.13}$$

  Here, $\mu_{ij} = \frac{1}{E}\sum_{i,j}\frac{G_1(i,j)}{G_2(i,j)}$ is the mean ratio. The indicator gives a measure of the standard deviation of local changes in terms of contrast.

**Results**

As the images for benchmarks we use some synthetic images from Brodatz textures collection [8], besides the well known Baboon and Boat (Fig. 3.17). They are 512×512 in size but d72 (640 × 640). From left to right we show the original images, those specified using our algorithm to match as target PDF respectively a linear, Gaussian and logarithmic distribution (shown in the last row of Figure 3.17). We do not show images achieved with the other approaches, since the differences are not visually perceptible. Rather, they can be assessed through analyzing Tables 3.2 3.1 3.3, column by column.



**(a)** Original **(b)** Linear **(c)** Gaussian **(d)** Log

**Figure 3.17:** Brodatz d72 (top), Baboon (middle) and Boat (bottom) histogram specified using the proposed method with synthetic target histograms.

**Table 3.1:** Histogram specification results for image d72, $640 \times 640$, single channel

| Method | Speed | Distortion $\Delta$ | | | Match $D_{KS}[10^{-2}]$ | | |
|---|---|---|---|---|---|---|---|
| | | Lin | Gau | Log | Lin | Gau | Log |
| Classic | 56.58 | .023 | .244 | .147 | 4.07 | 4.15 | 4.39 |
| Eram$_\alpha$ | 1.88 | .024 | .559 | .198 | .743 | .975 | 1.04 |
| Eram$_\beta$ | 1.05 | .024 | .701 | .203 | .067 | .059 | .107 |
| Coltuc | 0.20 | .024 | .741 | .210 | .000 | .000 | .000 |
| Ours | 42.91 | .024 | .741 | .210 | .000 | .000 | .000 |

At a glance, we can see how the performance delivered by the proposed approach is identical to Coltuc, but for speed. In fact, the algorithm always performs far better than all the other ones (more that one order of magnitude, more than two as Coltuc is concerned) but the classic, whose speed is slightly higher. At the opposite, Coltuc is the slowest one.

As for image and histogram distortion indicators, results in the tables show they are inversely proportional, as might be expected: the lower the histogram distortion, the higher the image distortion. As far as image distortion is concerned, the best values are achieved using classic approaches, although in the linear distribution case performance are very close for every algorithm and image. On the contrary, the standard algorithm

**Table 3.2:** Histogram specification results for Boat, $512 \times 512$, single channel

| Method | Speed | Distortion $\Delta$ | | | Match $D_{KS}[10^{-2}]$ | | |
|---|---|---|---|---|---|---|---|
| | | Lin | Gau | Log | Lin | Gau | Log |
| Classic | 87.32 | .009 | .203 | .090 | 2.15 | 2.04 | 2.05 |
| $\text{Eram}_{\alpha}$ | 2.99 | .009 | .226 | .098 | .490 | .421 | .638 |
| $\text{Eram}_{\beta}$ | 1.56 | .009 | .250 | .099 | .027 | .026 | .028 |
| Coltuc | 0.36 | .009 | .249 | .103 | .000 | .000 | .000 |
| Ours | 71.14 | .009 | .249 | .103 | .000 | .000 | .000 |

**Table 3.3:** Histogram specification results for Baboon, $512 \times 512$, single channel

| Method | Speed | Distortion $\Delta$ | | | Match $D_{KS}[10^{-2}]$ | | |
|---|---|---|---|---|---|---|---|
| | | Lin | Gau | Log | Lin | Gau | Log |
| Classic | 88.82 | .010 | 1.08 | .324 | .766 | .788 | .786 |
| $\text{Eram}_{\alpha}$ | 2.99 | .010 | 1.09 | .339 | .134 | .204 | .198 |
| $\text{Eram}_{\beta}$ | 1.56 | .010 | 1.11 | .345 | .013 | .013 | .014 |
| Coltuc | 0.35 | .011 | 1.12 | .346 | .000 | .000 | .000 |
| Ours | 67.45 | .011 | 1.12 | .346 | .000 | .000 | .000 |

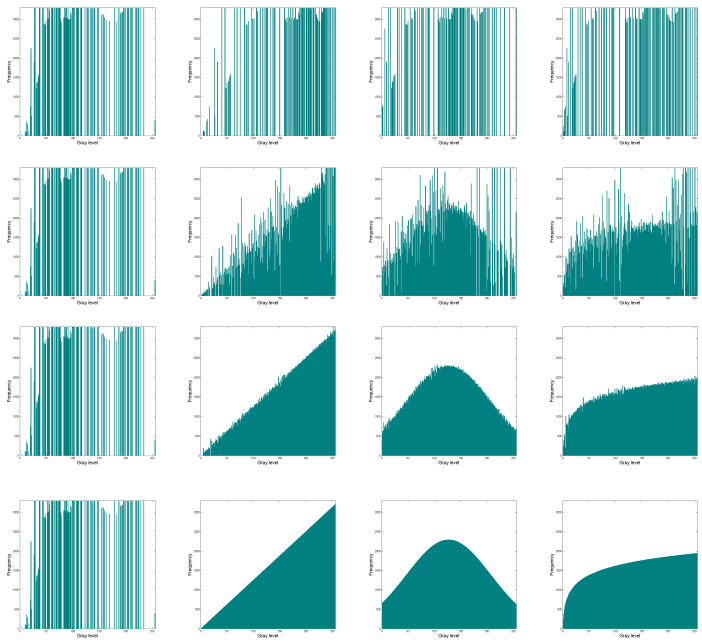
shows the worst histogram distortion .

Coltuc and the proposed method deliver the same image distortion and are the only algorithms to produce perfectly matching histograms, i.e. with no histogram distortion at all. As for histogram distortion, the best algorithm among quasi-exact methods is Eramian, that always shares with Coltuc and the proposed one comparable image distortion for all images. However, in any case, it never reach zero histogram distortion.

### 3.2.5 Conclusion

A novel method to perform a fast and exact histogram specification given a source image and a target histogram has been detailed. Usually, mapping between source and target histograms is described via analytic functions or rank statistics computed on the distribution of pixels brightness. However, histogram distortions artifacts such as gaps and overfull bins prevent to achieve exact histogram matching.

Additional features, such as neighborhood average brightness, have been introduced to discriminate among pixel having same brightness values, since indexing uniquely every single pixels would lead to exact histogram matching. Though, these methods



**(a)** Original **(b)** Linear **(c)** Gaussian **(d)** Log

**Figure 3.18:** Comparison between source (left column) and specified histograms using five different specification methods: from top to bottom, Classical, Eram$_\alpha$, Eram$_\beta$, ours (Coltuc's is identical and it has not been reported), and different target histograms: from left to right, linear, gaussian and logarithmic.

call for computationally expensive implementations.

Our approach achieves exact matching by replacing the standard mapping function with the concept of one-to-many relationship. This enables to spread undistinguishable pixels, i.e. having same brightness, to diverse target brightness values and avoids histogram distortion artifacts.

Established performance indicators have been used to assess quality and computational cost of the conceived algorithm. Results confirm that the proposed method runs more than two order of magnitude faster than the exact method and more than one order faster if compared with other quasi-exact approaches. This speedup has been achieved while maintaining comparable image distortion.

# Bibliography


[1] P. Azzari and A. Bevilacqua. Joint spatial and tonal alignment for motion detection with ptz camera. In *Proc. of Intl. Conf on Image Analysis and Recognition*, volume 4142, pages 764–775, 2006.

[2] A. Bartoli, N. Dalal, B. Bose, and R. Horaud. From video sequences to motion panoramas. In *Proc. of Workshop on Motion and Video Computing*, pages 201–207, December 2002.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[4] A. Bevilacqua and P. Azzari. A high performance exact histogram specification algorithm. In *Proc. of IEEE Intl. Conf. on Image Analysis and Processing*, pages 623–628, 2007.

[5] A. Bevilacqua, L. Di Stefano, and P. Azzari. An effective real-time mosaicing algorithm apt to detect motion through background subtraction using a ptz camera. In *Proc. of IEEE Intl. Conf. on Advanced Video and Signal based Surveillance*, volume 1, pages 511–516, 2005.

[6] A. Bevilacqua, L. Di Stefano, and A. Lanza. An efficient motion detection algorithm based on a statistical non parametric noise model. In *Proc. of IEEE Intl. Conf. on Image Processing*, pages 2347–2350, October 2004.

[7] K. S. Bhat, M. Saptharishi, and P. K. Khosla. Motion detection and segmentation using image mosaics. In *Proc. of Intl. Conf. on Multimedia and Expo*, volume 3, pages 1577–1580, 2000.

[8] P. Brodatz. Textures: a photographic album for artists and designers. In *Dover Publications*, 1999.

[9] M. Brown and D. G. Lowe. Recognising panoramas. In *Proc. of the IEEE Intl. Conf. on Computer Vision*, pages 1218–1225, 2003.

[10] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. on Graphics*, 2:217–236, October 1983.

[11] F. M. Candocia. Jointly registering images in domain and range by piecewise linear comparametric analysis. *IEEE Trans. on Image Processing*, 12(4):409–419, 2003.

[12] D. P. Capel. *Image mosaicing and super-resolution*. University of Oxford, 2001.

[13] D. Coltuc, P. Bolon, and J. M. Chassery. Exact histogram specification. *Trans. on Image Processing*, 15(5):1143–1152, May 2006.

[14] Intel©Corp. Opencv 1.0, open source computer vision library, 2000-2007.

[15] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, October 2003.

[16] R. Cucchiara, C. Grana, A. Prati, and S. Sirotti. Improving shadow suppression in moving object detection with hsv color information. In *Proc. of Intelligent Transportation Systems Conference*, pages 360–365, 2001.

[17] A. Eden, M. Uyttendaele, and R. Szeliski. Seamless image stitching of scenes with large motions and exposure differences. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 2498–2505, 2006.

[18] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density for visual surveillance. In *Proc. of the IEEE*, volume 90, pages 1151–1163, July 2002.

[19] M. Eramian and D. Mould. Histogram equalization using neighborhood metrics. In *Proc. of Canadian Conf. on Computer and Robot Vision*, pages 397–404, 2005.

[20] R. C. Gonzales and R. E. Woods. Digital image processing. *Upper Saddle River, NJ, Prentice Hall*, 2002.

[21] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*. Addison-Wesley, 2002. GON r 02:1 1.Ex.

[22] M. D. Grossberg and S. K. Nayar. Determining the camera response from images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(11):1455–1467, November 2003.

[23] M. Grundland and N. A. Dogson. Color histogram specification by histogram warping. In *Proc. of SPIE Color Imaging X: Processing, Hardcopy, and Applications*, pages 610–621, January 2005.

[24] E.L. Hall. Almost uniform distributions for computer image enhancement. *IEEE Transactions on Computers*, 23(2):207–208, February 1974.

[25] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of Alvey Vision Conf,*, pages 147–151, 1988.

[26] D. Hasler and S. Susstrunk. Colour handling in panoramic photography. In *Videometrics and Optical Methods for 3D Shape Measurements*, pages 62–72, January 2002.

[27] E. Hayman and J. Eklundh. Statistical background subtraction for a mobile observer. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 67–74, 2003.

[28] B. Brumitt K. Toyama, J. Krumm and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Proc. of Intl. Conf. on Computer Vision*, pages 255–261, 1999.

[29] S. Kang, J. Paik, A. Koschan, B. Abidi, and M. A. Abidi. Real-time video tracking using ptz cameras. In *Proc. of Intl. Conf. on Quality Control by Artificial Vision*, pages 103–111, May 2003.

[30] S. J. Kim and M. Pollefeys. Radiometric self-alignment of image sequences. In *Proc. of IEEE Intl. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 645–651, 2004.

[31] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60(2):91–110, November 2004.

[32] F. Pitié, A. Kokaram, and R. Dahyot. Towards automated colour grading. In *Proc. of IEEE European Conference on Visual Media Production*, London, November 2005.

[33] F. Pitié, A. Kokaram, and R. Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, February 2007.

[34] A. Rosenfeld and A. Kak. Digital picture processing. *Upper Saddle River, NJ, Prentice Hall*, 1982.

[35] M. Saptharishi, K. Bhat, C. Diehl, C. Oliver, M. Savvides, A. Soto, J. Dolan, and P. Khosla. Recent advances in distributed collaborative surveillance. In *Proc. of SPIE on Unattended Ground Sensor Technologies and Applications*, pages 199–208, April 2000.

[36] H. S. Sawnhey and R. Kumar. True multi-image alignment and its applications to mosaicing and lens distortion correction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(3):235–243, March 1999.

[37] L.G. Shapiro and G.C. Stockman. *Computer Vision*. Prentice Hall Inc., New Jersey, 2001.

[38] Y. Sugaya and K. Kanatani. Extracting moving objects from a moving camera video sequence. In *Proc. of Symposium on Sensing via Image Information*, pages 279–284, June 2004.

[39] R. Szeliski. Video mosaics for virtual environments. *Computer Graphics and Applications*, 16(2):22–30, 1996.

[40] C. Tomasi and J. Shi. Good features to track. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[41] F. Winkelman and I. Patras. Online globally consistent mosaicing using an efficient representation. In *Proc. IEEE Intl. Conf. on Systems, Man and Cybernetics*, pages 3116–3121, October 2004.

[42] Y. Xiong and K. Turkowski. Registration, calibration and blending in creating high quality panoramas. In *Proc. of IEEE Workshop on Applications of Computer Vision*, pages 69–74, 1998.

[43] Y. J. Zhang. Improving the accuracy of direct histogram specification. *Journal of Electronic Imaging*, 28(3):213–214, 1992.

[44] Z. Zhu, G. Xu, E. M. Riseman, and A. R. Hanson. Fast generation of dynamic and multi-resolution 360 panorama from video sequences. In *Proc. IEEE of Intl. Conf. on Multimedia Computing and Systems*, pages 400–406, July 1999.

[45] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.

[46] S. Zokai and G. Wolberg. Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. *IEEE Trans. on Image Processing*, 14(10):1422–1434, October 2005.

# Chapter 4

# Evaluation methodology for image mosaicing algorithms

As soon as image mosaicing has been recognized as a key building block of many computer vision applications, the need for a principled and widespread methodology allowing to assess and compare the performance delivered by different approaches has become of primary importance. Indeed, several image mosaicing algorithms claiming to advance the state of the art have been proposed in recent years. Though, improvements can be sometimes recognized without quantitative evidences, a quantitative methodology for comparing different algorithms is essential as this discipline evolves.

What algorithm is the best? How to ascertain its primacy? To answer such questions, this section proposes a comprehensive evaluation methodology including standard data sets, ground-truth information and performance metrics. Aside the explanation of the key components, the performance of three variants of a well-known mosaicing algorithm are evaluated according to the proposed methodology.

## 4.1 Introdution and related work

Image mosaicing represents a popular way of achieving a dense scene reconstruction by composing several overlapping views of the same scene matter. It can be regarded as a special case of scene reconstruction when the images are spatially related by a planar collineation (homography) or subclasses of this transformation (affinity, similarity, translation). As pointed out in section 3.1.2, this assumption holds when images exhibit no parallax effects, i.e. when the scene is approximately planar or the camera purely rotates about its optical center. In these circumstances, knowledge of the planar geometric transformations among images permits to reconstruct a dense model of the

scene, known also as mosaic or panorama.

Several mosaicing algorithms aimed at advancing the state-of-the-art have been proposed in literature. Some innovations such as the topology inference proposed by Shawney [13], the global geometric consistency proposed by Shum [15] or the recent automatic panorama recognition presented by Brown [5] clearly provide sharp improvements over the existing state of the art. However, this is not always the case and due to the lack of a reference test bed it is often very difficult, or even impossible, to evaluate and compare different mosaicing algorithms. Moreover, only visual inspections or problem specific metrics have been used so far for performance assessment. The adoption of metrics based on human perception arises from the fact that in the past mosaics have been mostly used in computer graphic applications aimed to a human audience, such as publicity, photomontage, special effects.

Nowadays mosaicing algorithms are employed not only to generate visually pleasant pictures but also serve as key building blocks for many computer vision applications, such as e.g. motion detection and tracking [3, 9], mosaic-based localization [10], resolution enhancement [6], augmented reality [1]. In such scenarios, visually similar mosaics can be characterized by different levels of numerical accuracy and hence have a different impact on the addressed computer vision applications.

We believe that in these settings a proper reference test bed and evaluation methodology is needed, so as to allow for quantitative performance assessment. Moreover, algorithms are becoming so accurate that human based perception metrics will soon be unable to meaningfully distinguish mosaics obtained with different algorithms (e.g. the mosaics in the left column of Fig.4.1 look identical but they turned out very different in terms of accuracy of reconstruction of the original scene, see Fig.4.2).

Inspired by the renowned work of Scharstein [14] and the more recent work by Baker [2], respectively in the field of stereo matching and optical flow, this section proposes an evaluation methodology for mosaicing algorithms that allows for principled quantitative discussion about performances and represents a useful tool for other researchers. The proposed methodology enables to rate any mosaicing algorithm based solely on the output yielded on standard data sets, and therefore irrespectively of any knowledge on its theoretical foundations or implementation. To this purpose, we have conceived a framework consisting of data sets and tools for their creation, ground-truth information and performance metrics. As a case study, the methodology has been applied to the comparison and ranking of three variants of a well-known mosaicing algorithm that produce high quality, as well as visually indiscernible results.

To the best of our knowledge, there exists no other similar performance evaluation framework in the field of image mosaicing. The issue of performance evaluation is addressed in two well known references [4, 16] that are thorough surveys of the literature

in the field of planar image registration. Although covering a wide range of algorithms and applications, the suggested performance indicators pertain only to specific classes of methods, e.g. keypoints-based algorithms, and may not be widely applicable.

An on-line version of our results, along with the data sets with ground-truth used in this work, can be found at: `http://www.vision.deis.unibo.it/MosPerf`. This web page includes also an online form that allows researchers to download the data sets and then submit their own results for evaluation.

## 4.2 Evaluation methodology

Quantitative evaluation has been usually achieved by calculating errors statistics among registered images of the input sequence. This corresponds to the adoption, within a mosaicing framework, of performance metrics borrowed from image registration theory. Examples of such performance indicators can be found in [4, 16] These indicators require a set of corresponding control points to be available, so as to compute error statistics, e.g. the mean square distance, between the image data and the predictions yielded by the algorithm at hand. However, this approach suffers from at least four major drawbacks:

- comparison among different algorithms is impossible unless the very same set of control points is used. To the best of our knowledge such a reference test bed has not been proposed so far.

- an algorithm cannot be evaluated based solely on its output, since the registration transformations need to be available to compute error statistics.

- any set of control points can be exactly fit using a sufficiently highly parameterized registration model (overfitting), thus defying these statistics

- algorithm accuracy and noise affecting the data are coupled, error statistics can take large values even in case of good fitting only because of noisy measurements.

Instead, the proposed quantitative evaluation methodology relies on the computation of error statistics obtained by comparing the mosaic yielded by a given algorithm, on a reference data set (i.e. a sequence of images to be stitched together), to the corresponding ground-truth mosaic (i.e. the mosaic that would be obtained by exactly stitching together the images of the reference data set). To the best of our knowledge there exists no work proposing a quantitative evaluation methodology for mosaicing algorithms based on comparison with ground-truth information.

The approach outlined in this section holds the potential to allow for fair and informed quantitative evaluation of algorithms based solely on their outputs. This is a very important point: since the comparison is taken to a higher level of abstraction. The proposed framework does not require the algorithms to use control points nor homographic registration models. We only assume that the ”algorithm” accepts several images as input for creating a composite image, no matter whether it be a software running on a laptop, an hardware implementation or just a skilled photographer. As a matter of fact, a crucial ingredient in our proposal is the availability of reference data sets with accurate ground truth. How to obtain such data? The issue is addressed in the next section.

### 4.2.1 Generation of data sets with ground truth

We focus here on the method used to collect data sets with ground-truth and defer the selection of specific data sets to Section 4.3. The data sets generation problem can be approached from two main directions:

- acquisition of real measurements using alternative methods that ensure a much higher degree of precision compared to that affordable by the techniques under assessment. For example, authors in [14] used structured-light to obtain highly reliable ground truth. Indeed, the advantage of this method is the generation of data sets consisting of real-world data and real challenges. On the other hand care must be taken to ensure that the ground-truth method is really accurate and unbiased. Moreover, the controllability of the test bed environment remains an important issue. Is it manageable to collect several data sets each of them isolating a single peculiar aspect such as different degree of optical distortion, different light conditions while maintaining everything else roughly constant?

- creation of synthetic data that bear good resemblance with real imagery, for example by rendering detailed scenes using a computer graphics environment. From this vantage point, the computed imagery will always be somehow synthetic but the controllability is complete. Unfortunately, general purpose renderers such as PoV [12] have been mostly conceived for computer graphics applications and some computer vision aspects are not easily embeddable in this framework. Are radiosity and photon mapping algorithms really important if non ideal optical lenses need still to be simulated with a custom postprocessing stage? Not to mention non linear camera response function or sensor noise.

In the end, both approaches are interesting on their own and can be tweaked to emphasize different challenges that a mosaicing algorithm must be able to tackle. Nonetheless, there is a third intermediate way envisioned by authors in [2], through

which they claimed to obtain "realistic synthetic imagery" using image interpolation techniques and computer graphics tools. Along the same lines, we have developed a software component, called Virtual Camera (VC) that generates photorealistic synthetic images using a mixture of real and precomputed information. By exploiting the geometry of projective planes, the VC approach retains both controllability and realism while being easy to implement and computationally cheap.

Controllability descends from the fact that VC simulates the geometric image formation process of today imaging devices taking into accounts internal parameters, pose and position, sensor size and resolution, focal length and sensor noise. Simplicity comes from the fact that the actual scene is just a plane. This does not represent a loss of generality since the constraint of lack of parallax required to properly apply planar registration techniques is naturally enforced in this way. The realism comes from the fact that a real picture is used to texture the planar scene framed by the VC. In this way realistic noise is naturally embedded in the framework and need not to be simulated using synthetic statistical distributions.

Hence, VC is a fully configurable renderer able to generate images of a realistic planar virtual scene. Moreover, any virtual frame can be easily created by just defining a simple homography $H$, as explained in the remainder of this section. Denoting a $2D$ point as $x = [u, v]$ and a $3D$ point as $X = [X, Y, Z]$, Eq. 2.4 relates a $3D$ point $X$ and its projection on the image $x$.

Since the scene model is a plane, we can assume, without loss of generality, that it is located on $Z = 0$ of the world coordinate system. Denoting the $i^{th}$ column of the rotation matrix $R$ by $r_i$, from eq. 2.4 follows

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix}\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K\begin{bmatrix} r_1 & r_2 & t \end{bmatrix}\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{4.1}$$

By still using $X$ to denote a point on the scene plane, even though $X = [X, Y]$ since $Z$ is always equal to 0, a scene point $X$ and its image projection $x$ are related by a homography $H$ given by

$$s\widetilde{x} = H\widetilde{X} \text{ with } H = K\begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \tag{4.2}$$

Hence, to collect a data sets sequence, a reference image is initially chosen (i.e. a satellite or aerial image) and then a list of VC parameters, one for each snapshot, is computed. These parameters encode the desired trajectory and internals of the camera. In this manner, different positions and orientations are used to generate the translation and panning sequences of the actual datasets. Every snapshot of the sequence is just

the projection of the scene onto the virtual camera sensor according to Eq. 4.2 and the VC parameters.

The ground truth mosaic is simply generated by cutting-and-pasting the portion of the reference image that has been viewed by the VC during the sequence (i.e. a pixel of the reference image belongs to the ground-truth mosaic if it has been projected in at least one snapshot of the data set). Due to its simplicity, this approach ensures that the ground truth is completely unbiased and does not favor any conceivable method.

Several issues must be careful considered in order to generate meaningful data sets. The most important is the pixelation effect. The pixelation effect is known in computer graphic as the artifact that causes individual pixel to be visible to the eye, mostly because the image has a lower resolution than the medium is being displayed on. In these scenario the pixelation effect can occur because the camera is too slanted or gets too close to the scene plane, so that texture projection requires oversampling. To avoid this undesirable artifact, a minimum distance and a maximum rotation of the VC with respect to the scene, given the texture resolution, are estimated beforehand and used as thresholds.

A very similar workaround has been adopted to avoid strongly deformed mosaics that would require image oversampling during the reconstruction stage. All the images comprising a sequence are taken so that they are compliant with the aforementioned threshold.

### 4.2.2 Data normalization

Some relevant issues concerning the normalization of the delivered mosaics must be properly taken into account, in order to be able to compare different algorithms based solely on their outputs.

Registering a sequence of $N$ views amounts at finding the $N \times N$ pairwise transformation $H_{i,j}$ that links each view to another. As discussed in Section 3.1.2, using graph theory this can be seen as a view-graph with images being nodes and transformations being edges connecting nodes. In this settings, we would end up with a huge $K_N$ complete graph and a terrific computational cost. However, most of the transformations are not independent since to be compatible they must fulfill the condition that a composite transformation computed by concatenation around any cycle in the view-graph is equal to the identity.

Thus only a subset of $(N-1)$ transformations touring an arbitrary maximal cycle is required to completely describe the problem. In addition, since the view order is unimportant, an arbitrary order can be induced in the sequence, obtaining a transformation chain $C$ where the individual transformations could be written in the form $H_{i-1,i}$ with $i \in [1..N-1]$. For this reason, two registration algorithms $A, A'$ are equivalent if their

transformation chains $C, C'$ are the same:

$$H_{i-1,i} = H'_{i-1,i},\ i \in [1..N-1] \tag{4.3}$$

Once the homography chain $C$ is known, the creation of the mosaic requires to fix a coordinate frame, refereed to here as the reprojection coordinate system (RCS), through the choice of a rendering matrix $R_0$ applied to a reference frame $I_0$. Once $R_0$ has been fixed, the visualization matrices $Q_i$ through which every image reprojects in the RCS takes the form

$$Q_i = R_0 \prod_{j=1}^{i} H_{j-1,j},\ i \in [0..N-1] \tag{4.4}$$

The reference frame is not special, for the very same mosaic could be obtained by selecting any other frame $I_i$ in the sequence and computing the visualization matrices $Q_i$ accordingly.

The RCS is usually chosen as the coordinate system of one image in the sequence, so that the rendering matrix would be the identity for that image. In other cases, the choice may be driven by another criterion, e.g. minimum global distortion of the panorama. The rendering matrix $R_0$ (typically a translation and a scale change, but, in principle, even a homography) links the RCS to an arbitrary reference image of the sequence.

When comparing two panoramas built from the composition of images warped according to homography chains, one can try to compare corresponding pixels of the two images. For these reason it can be stated that, two registration algorithms $A, A'$ produce equivalent mosaics if the corresponding visualization matrices are all the same

$$Q_i = R_0 \prod_{j=1}^{i} H_{j-1,j} = R'_0 \prod_{j=1}^{i} H_{j-1,j}{}' = Q'_i,\ i \in [1..N-1] \tag{4.5}$$

Since we cannot expect the rendering matrices $R_0, R'_0$ chosen by different algorithms to be the same, the resulting mosaics will exhibit different corresponding pixels even if homography chains are identical, and thus contradicting the definition of equivalent registration algorithms. In other terms, the concept of equivalent registration does not imply the concept of equivalent visualization except for the case $R_0 = R'_0$

Therefore, since we want to appraise the registration capabilities of mosaicing algorithms by analysing the delivered mosaics, a major issue to be dealt with before the computation of the performance metrics is normalization of the panoramas. This amounts at filtering out the visualization effects due to different choices of the rendering matrix $R_0$ so that all panoramas will lay in the same RCS even though originally built in different rendering coordinate systems. By doing that, the remaining discrepancies between the panoramas will be due to registration inaccuracies, i.e. different registration matrices along the homography chains.

This is the reason why an $R_0$ default rendering matrix and a corresponding reference frame, i.e. the first of the sequence, are specified for every sequence of our data sets. By imposing these two additional constraints, it is ensured that any algorithm will render in the same RCS as that of the ground-truth mosaic. Thus, since the ground-truth mosaics and those generated by the algorithms are normalized, performance metrics based on the comparison of corresponding pixels become appropriate.

Finally, it is worth pointing out that since the frames forming data set sequences are generated according to known homograpies (i.e. by Eq. 4.2), it is also possibile to render a panorama using these known trasformations and $R_0$, $I_0$. Such an image would not be affected by registration errors, for the homography chain being exactly known, and hence differ from the ground truth mosaic only because of the effects of the resampling and interpolation processes. The performance metrics associated with the panoramas rendered using the known transformations will be reported in Section 4.3, as they can be seen as upper bounds on the performance attainable by mosaicing algorithms.

### 4.2.3 Performance metrics

As mentioned in the previous section, provided that data are properly normalized, different algorithms can be assessed and ranked based on direct pixelwise comparison between the generated and the ground truth mosaics. Denoting the mosaic under evaluation as $I_C$ and the ground truth as $I_T$, the following performance metrics have been defined:

1. Average of the intensity distances. It amounts to the MSE over intensities of corresponding pixels

$$\text{MSE} = \frac{1}{M}\sum_{(x,y)} D_{xy} = \frac{1}{M}\sum_{(x,y)} \left(m_C(x,y) - m_T(x,y)\right)^2 \tag{4.6}$$

   where $(m_C(x,y), m_T(x,y))$ are corresponding pixels in $I_C$, $I_T$ and $M$ is the number of pixel belonging to the region of overlap between the two images. Pixels not shared by both images are neglected.

2. Average of the geometric distances. It amounts to the MSE of the distances between corresponding control points in $I_C$, $I_T$

$$\epsilon_{est} = \frac{1}{L}\sum_{i} D_i = \frac{1}{L}\sum_{i} \left\|(x_C^i, y_C^i) - (x_T^i, y_T^i)\right\|^2 \tag{4.7}$$

   where $L$ is the number of correspondences. Corresponding control points $(x_T^i, y_T^i) \rightarrow (x_C^i, y_C^i)$ are obtained by matching $L$ KLT keypoints, located over an approximately regular grid, between $I_T$ and $I_C$.

| Method | PT | | | | PR | | | | LP | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | Mis | $\epsilon_{est}$ | Time | MSE | Mis | $\epsilon_{est}$ | Time | MSE | Mis | $\epsilon_{est}$ | Time |
| SR-KLT | **226.98** | **0.092** | **0.098** | 1.17 | 54.71 | 2.686 | 0.561 | 1.49 | **606.47** | **1.203** | **0.238** | 3.34 |
| SR-Harris | 231.67 | 0.645 | 0.143 | 1.14 | 51.25 | **1.431** | 0.471 | 1.45 | 756.49 | 1.975 | 0.436 | 3.22 |
| SR-SIFT | 279.80 | 2.395 | 0.381 | 26.41 | **48.71** | 1.648 | **0.363** | 9.72 | 1106.23 | 2.982 | 0.675 | 54.62 |
| SR-GT | 223.62 | 0 | 0.093 | | 47.85 | 0 | 0.306 | | 536.71 | 0 | 0.120 | |

**Table 4.1:** Experimental results on sequences PT, PR and LP.

3. Number of misplaced pixels. It is the sum of missing and redundant pixels normalized with respect to $N$

$$\text{Mis} = \frac{1}{N}(R+P) = \frac{1}{N}\Big(\sum_{(x,y)}((x,y) \in m_C \wedge (x,y) \notin m_T) + \sum_{(x,y)}((x,y) \in m_T \wedge (x,y) \notin m_C)\Big) \tag{4.8}$$

Since Mis is often a very small number, it has been scaled by $10^3$ in tables 4.1 and 4.2 of next section.

## 4.3 Experimental results

This section aims at comparing three mosaicing algorithms on the basis of the proposed methodology.

The algorithms are iterative variants of the well known Direct Linear Transform (DLT) registration algorithm [7]. The DLT algorithm estimates the spatial transformation occurring between two images (pairwise registration) performing a linear regression on a set of corresponding points. The transformation model is an over-parameterized 9 dof homography and the system is solved using Singular Value Decomposition (SVD). Robust estimation is obtained performing outliers removal with the RANSAC algorithm. The mosaicing algorithm is an iterated application of this registration algorithm along pair of frames of the sequence. Sequential concatenation of $n$ pairwise registrations amounts at finding the transformation that relates the $n^{th}$ view to the reference one and thus to the RCS.

The three algorithms differ in the features detection and tracking methods employed to determine the set of corresponding points. The first two algorithms, referred to as SR-Harris and SR-KLT (SR stands for Sequential Registration), rely on respectively the Harris and the KLT detector for features extraction. Both algorithms match detected features by means of the KLT tracker. Since this kind of tracker suffers from large

| Method | PTEx | | | | LPEx | | | |
|---|---|---|---|---|---|---|---|---|
| | MSE | Mis | $\epsilon_{est}$ | Time | MSE | Mis | $\epsilon_{est}$ | Time |
| SR-KLT | **466.43** | 2.277 | **0.390** | 4.99 | **715.48** | **1.774** | **0.378** | 8.77 |
| SR-Harris | 574.55 | **1.988** | 0.490 | 4.84 | 850.88 | 3.333 | 0.538 | 8.69 |
| SR-SIFT | 895.75 | 7.883 | 0.791 | 143.63 | 1279.22 | 5.636 | 0.741 | 89.86 |
| SR-GT | 218.23 | 0 | 0.096 | | 520.47 | 0 | 0.119 | |

**Table 4.2:** Experimental results on extended sequences PTEx and LPEx.

shift, its robustness has been increased with a coarse initial guess by means of a phase correlation step. The third algorithm, referred to as SR-SIFT, uses the SIFT detection and tracking implementation described in [8]. The three algorithms share the same simple blending method; a simple pixelwise average of color values within overlapping areas has been chosen (see Section 3.1.2 for different approaches).

Each test sequence consists of a collection of views, a rendering matrix and a reference frame to which the supplied rendering matrix must be applied to identify the rendering coordinate system. According to the image formation model described in Section 4.2.1 the focus has been on sequences with spatial misalignments only., for the recovery of the spatial structure is the primary concern of most mosaicing algorithms known in literature.

The five sequences [1] are:

- Pure Translation (PT): it consists of 9 frames acquired by translating on the right and keeping the optical axis of the virtual camera orthogonal to the scene plane. Adjacent frames overlap by a 30% – 50% of their area and small vertical misalignments have been added.

- Pure Rotation (PR): it is composed of 9 frames acquired by rotating the virtual camera around the $Y$ axis ($Z$ pointing toward the observer). Adjacent frames are spaced by 4 degrees and overlap is about 80%.

- Looping Path (LP): it consists of 18 frames, acquired by moving the virtual camera on a loop by means of translations, above the $X, Y$ plane parallel to the scene, so that the last frame roughly overlaps the first frame.

- Pure Translation Extended (PTEx) and Looping Path Extended (LPEx) are longer sequences (36 and 37 frames respectively) that extend PT and PR by including, respectively, repeated panning and looping.

[1] Images used by the virtual camera are courtesy of NASA Earth Observatory [11]

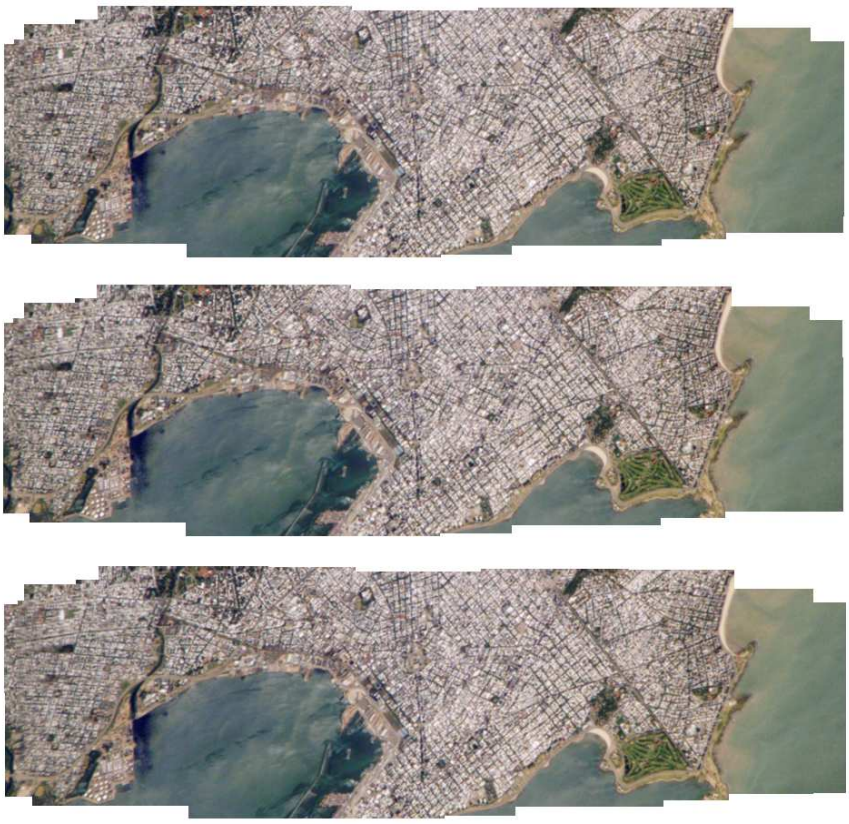


**Figure 4.1:** Mosaics generated from sequence Pure Translation. From top to bottom: SR-KLT, SR-Harris and SR-SIFT.

Two important remarks are worth to be emphasized:

- all the sequences do not feature illumination changes; this is a design choice taken to focus on the geometrical part of the mosaicing problem by decoupling it from photometric aspects.
- some of the sequences exhibit basic camera motions and might not be considered as representative of real world sequence. This is another design choice taken to dissect possible camera motion into several primitives and to study the performance of the algorithms on them independently.

Table 4.1 and Table 4.2 report for each algorithm and for each sequence the performance metrics MSE, Mis, $\epsilon_{est}$ and the execution time. SR-GT, reported in the last row of each table, refers to a pseudo-algorithm that composes the mosaic based on the known transformations used by VC to generate the data sets. For each performance

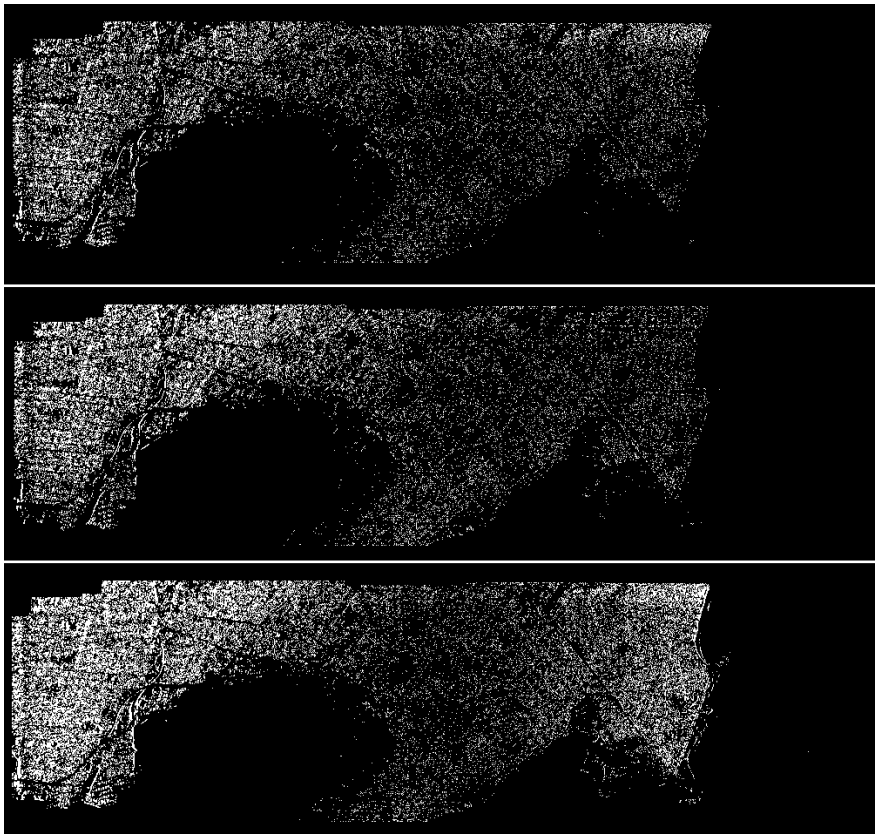


**Figure 4.2:** Sum of Squared Differences maps computed by subtracting generated mosaics and the ground truth. From top to bottom: truth SR-KLT, SR-Harris and SR-SIFT.

metric the best performing algorithm is highlighted in boldface.

Tables 4.1 and 4.2 show clearly that on the whole dataset, with the exception of sequence PR for which all the algorithms perform very close to SR-GT, SR-KLT is the best performing algorithm. Tables show also that overall, SR-Harris outperforms SR-SIFT. Notably, on the PR sequence SR-SIFT takes advantage of its rotation invariant features. This clear ranking is impressive if compared to the similar appearance of the three mosaics reported in Figure 4.1. On the contrary, the SSD (Sum of Squared Differences) maps depicted in Figure 4.2 (whose average value is the MSE performance metric) allow to appreciate the local differences between the mosaics.

An interesting remark stems from pairwise comparison of the performance of SR-KLT, SR-Harris and SR-SIFT on short and extended sequences (PT vs PTEx and LP vs LPEx). Even though the framed portion of the scene is substantially the same for both pairs, all the metrics agree on the fact that the longer the sequence the worst the mosaic, no matter the algorithm or the sequence. Such increasing inaccuracy is known

as drift error and manifest itself as the *looping path problem* [3], named after the fact it is visually emphasized in looping path sequences (that is, sequence that loops back so that the pair of images overlap after several frames). However, as pointed out by Tables 4.1 and 4.2 the drift accumulation is an inherent drawback of sequential algorithms, not depending on the sequence. Conversely, SR-GT exhibits an opposite behavior since the average of several corresponding pixels corrupted by resampling noise is a good estimate of the noise-free value. This suggests that the resampling error is normally distributed.

As a final remark, it is worth highlighting that the most suitable quality indicator when dealing with geometric misalignments only, as it is our case, is $\epsilon_{est}$. However, this not always applies since in the general case photometric changes occur as well. Under these circumstances, even a perfect spatial alignment ($\epsilon_{est} = 0$) could yield mosaics showing significant color differences compared to the ground truth. In general, the MSE measure, which senses both geometric and photometric alignment errors, is a more appropriate choice. These experiments show that MSE is monotonically related to the "exact" $\epsilon_{est}$ estimator, thus empirically validating the MSE metric as a quality measure of the mosaic.

## Conclusions

Image mosaicing techniques have a long history, evaluation methodologies for their comparison have not. Throughout this section a complete evaluation methodology including data sets, ground-truth information and performance metrics have been devised. The proposed data sets comprises 5 synthetic test sequences created by means of a fully configurable virtual camera. Simple pixelwise performance metrics such as the MSE have been employed to favor fairness and simplicity. The definition of a default visualization matrix and a reference frame is a simple procedure aimed at filtering out differences among mosaics visualized in different rendering coordinates system.

Afterwards, three variants of a known algorithm have been evaluated and compared according to the proposed methodology. Despite the fact that these approaches generates very good as well as visually similar results the evaluation procedure clearly shows that the KLT-based algorithm performs better.

In conclusion, we are firmly convinced that a widely accepted quantitative evaluation procedure is of utter importance as a branch of a discipline moves from its pioneering works to maturity. The purpose of this work has been to highlight this shortage and to propose an evaluation methodology that we hope will allow for principled discussion about algorithm performances and represent a useful tool for other researchers. Further information concerning the proposed evaluation methodology can be found at

the web site http://www.vision.deis.unibo.it/MosPerf.

Future developments are directed toward the creation of more challenging datasets featuring spatial as well as tonal misalignments, in the attempt of reduce the gap to synthetic realistic sequences. Moreover, the evaluation of more sophisticated algorithms, both through in-house development and direct collaboration with authors, is envisioned and promoted by the, currently under construction, on-line evaluation service hosted on the site.

# Bibliography


[1] P. Azzari, L. Di Stefano, F. Tombari, and S. Mattoccia. Markerless augmented reality using image mosaics. In *Proc. of Intl. Conf. on Image and Signal Processing*, 2008.

[2] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 1–8, October 2007.

[3] A. Bevilacqua and P. Azzari. High-quality real time motion detection using ptz cameras. In *Proc. of IEEE Intl. Conf. on Advanced Video and Signal based Surveillance*, page 23, 2006.

[4] L. Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[5] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *Intl. Journal of Computer Vision*, 74(1):59–73, August 2007.

[6] D. Capel and A. Zisserman. Computer vision applied to super resolution. *IEEE Signal Processing Magazine*, 20(3):75–86, May 2003.

[7] R. Hartley and A. Zisserman. *Multiple view Geometry in computer vision*. Cambridge University Press, Second Edition, 2003.

[8] R. Hess and A. Fern. Improved video registration using non-distinctive local image features. In *Proc. of IEEE Intl. Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[9] M. Irani, P. Anandan, J.R. Bergen, R. Kumar, and S. Hsu. Efficient representations of video sequences and their applications. *Signal Processing Image Communication*, 8(4):327–351, May 1996.

[10] Alonzo Kelly. Mobile robot localization from large-scale appearance mosaics. *Intl. Journal of Robotic Research*, 19(11):1104–1125, 2000.

[11] Nasa© Earth Observatory. Picture of the day gallery.

[12] PoV-Ray. Persistence of vision raytracer.

[13] H. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *Proc. of European Conf. on Compuer Vision*, pages 103–119, 1998.

[14] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Intl. Journal of Computer Vision*, 47(1-3):7–42, 2002.

[15] H. Shum and R. Szeliski. Systems and experiment paper: construction of panoramic image mosaics with global and local alignment. *Intl. Journal of Computer Vision*, 36(2):101–130, 2000.

[16] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.

# Chapter 5

# Camera pose reconstruction

Camera pose reconstruction addresses the problem of recovering the position and orientation, the pose, of a camera with respect to a given spatial coordinate. Visual pose reconstruction algorithms determine camera pose by relying only on information extracted from images. The camera pose reconstruction from image analysis in its general statement can be a tough problem, nonetheless a couple useful assumptions may be employed without harming generality too much. In particular, assuming previously calibrated cameras and presence of flat objects in the scene is realistic in many scenarios. In this settings, pose reconstruction can be cast back to a homography estimation problem, as anticipated in chapter 2.

In the next sections, two original applications, building on the concepts and algorithms of camera pose reconstruction, are illustrated. The first section is concerned with the proposal of an innovative use of image mosaics to boost the performance of known pose reconstruction algorithms. An augmented reality (AR) system, exploiting such mosaic-based pose reconstruction technique, has been implemented to demonstrate the improvements compared to conceptional approaches. The conceived AR system has been able to deliver real time, stable and realistic rendering of virtual objects and animations in several videos of real scenes.

The second section focuses on a novel human-machine interface concept for gaming applications based on visual camera pose reconstruction. In this context, a user interacts with the application by moving a hand held camera, the commands inferred from the reconstructed camera movements being conveyed as input to the videogame. Such a way of interacting ought to be practical and intuitive as long as 3D commands need to be naturally imparted to applications or electronic appliances. A proof of concept game has been also developed to demonstrate feasibility and effectiveness of the conceived vision-based interface.

## 5.1 Markerless augmented reality using image mosaics

Augmented reality aims at delivering spatially coherent information to a user moving in a known environment. Accurate and reliable pose estimation is the key to success. Many approaches track reference objects into the scene but as the environment becomes larger more objects need to be tracked. leading to computationally intensive methods. Instead, we propose an original approach that is suitable for environment where big planar structures are present. Several images of coplanar objects, or zoomed-in pictures of big planar structure, are composed into a large reference object using image mosaicing techniques, so that the pose reconstruction problem is simplified to that of finding the pose from a single plane. Experimental results show the effectiveness of this approach on two interesting case studies, i.e. aeronautical servicing and cultural heritage.

### 5.1.1 Introduction and related work

Augmented reality techniques convey information that is both semantically and spatially coherent with the observed scene. Information is shown by augmenting the scene captured through a camera with graphical objects that are properly aligned with the 3D structure of the scene and often contextually close to the user needs. In this section we mainly focus on structural coherence, nonetheless a simple demonstration of contextual awareness is given in the experimental results section.

The capability to deliver spatially coherent information to a user moving in a known environment is enabled by accurate and reliable pose reconstruction algorithms. Such algorithms try to compute the pose of the observer with respect to the world the user is moving in by establishing correspondences among objects detected in the scene. Based on these correspondences, both the information to be displayed and the structure of the scene is estimated.

Most of the algorithms described in literature can be thought of in terms of a binary taxonomy: those that rely on absolute information [22, 18], such as known models, and those based on chained transformations [23, 25]. The former seek to find camera poses that correctly reproject some fixed features of a given $3D$ model into the $2D$ images. They do not suffer from estimation drift but often lack precision, which results in jitter. The latter do not exploit a priori information but match interest points between images. Since correspondences between adjacent frames can be located precisely, usually these algorithms do not jitter but instead suffer from drift or even loss of track.

Pose estimation algorithms represent the world as a collection of reference objects, usually modeled as $3D$ meshes, associated with appearance models, such as collection of key frames or image patches related to each vertex. Navigation of large environ-

ments is handled using several objects spread across the scene, so that many of them are visible even though the user moves widely inside the environment. Many algorithms are known to estimate the pose very quickly using a single object and a single image [22, 18]. However, in presence of several objects, the pose of the observer is optimized together with the relative position of the visible objects typically using temporal coherence constraints, i.e. objects projections in different images are expected to confirm the same pose. As the environment grows larger so does the number of required objects, thus yielding to computationally intensive algorithms.

To reduce the complexity Simon et al. [23] and Uematsu et al. [25] considers only planar reference objects. In this settings they can exploit both temporal and spatial coherence in the estimation, i.e. homographies between planes can be computed independently and deployed as additional constraints. This involves constructing at each frames a unified projective space and mapping all the planes to that space according to computed homographies. The pose is subsequently calculated using correspondences between the space and image projections.

Nonetheless when several planar reference objects are also coplanar, the unified projective space can be profitably built in advance using image mosaicing techniques. As the cluster of objects becomes larger, using a mosaic as appearance model instead of a single shot, taken from larger distance or with shorter focal length, becomes more and more useful. In fact, the mosaic approach allows to maintain plenty of details that a single shot would miss.

We propose a practical approach that is suitable for environment where big planar structures are present. By mosaicing images of several coplanar objects, or zoomed-in pictures of a big flat structure, during a training stage, most part of the computation required to recovery the pose is shifted off-line. At run-time, the algorithm simply determines the pose with respect to a unique large reference object using approaches, such as [22, 18, 24], that are known to be fast and robust. This notably diminishes the on-line computational requirements and increases the accuracy of the estimated pose.

### 5.1.2 Methodology

The method is split up into two distinct stages. The first can be regarded as a training phase and is performed off-line. It deals with the definition of a large planar reference object together with the construction of its appearance model, i.e. a mosaic of images that portray the planar structure. Several keypoints are extracted from the appearance model using the SIFT features detector [15]. Metric measurements can be easily introduced in this framework by specifying the real world position of at least four non collinear points within the planar objects and computing the metric to projective homography accordingly.

The second stage performs on-line and addresses the estimation of the pose of the observer at a given instant using a set of points correspondences between the visible scene and the constructed appearance model. This stage encompasses a feature tracker, that establishes keypoint matches, and may deploy any pose estimation algorithm based on point correspondences. The projection of virtual objects is easily accomplished once the pose is known.

**Construction of the appearance model**

The first stage concerns the construction of the large reference object and its appearance model from a collection of pictures using a mosaicing algorithm. The idea of using mosaics in augmented reality applications is not a novelty in itself. For instance, Dehais et al. [5] use mosaics to augment the scene with virtual objects. However, with their system the user is allowed to rotate only and both the training and the testing sequence must be captured from the same vantage point. The approach proposed by Liu et al. [14] is also based on image mosaicing, but it requires fiducial markers and the viewpoint is again allowed to rotate only. Instead, our method relies on natural markers present in the scene and allows for arbitrary motion as long as a sufficient portion of the model is visible to the observer.

During a training stage the construction of the appearance model using several views of a roughly planar structure in the scene is carried out. The transformations among the views are homographies as long as the observed subject is planar. The algorithm we use to mosaic images can be regarded as an iterative version of the pair-wise DLT method described in [10] and evaluated in Chapter 4. From each pair of views a set of point correspondences is established and the best homography $H_{i,j}$ in the least square sense is fit; then the procedure is repeated for all pairs and visualization matrices $Q_i$ are computed. The rendering coordinate systems onto which images are composed into the mosaic turns out to be the common projective space computed by [25], provided that all patterns are coplanar.

Instead of building a mosaic, one might also capture the whole planar structure with a single shot taken from a larger distance or with a shorter focal length and then use such a shot as the appearance model. Indeed, this choice is potentially preferable when, given the resolution of the acquisition device, objects are as small as they can be captured by a single shot without losing too much information. In fact, in such a case objects are already registered with respect to each other and taking a picture is quicker than building a mosaic. Indeed, in any application scenario the more appropriate approach should be identified carefully. In the experimental results section, a comparison between the two approaches, in two different case studies, is presented.

Finally, given the appearance model, the SIFT feature detector extracts a set of

keypoints $x_i$ from it. Extracted features that appear in the model but do not belong to the planar reference object are discarded using a homography-based RANSAC algorithm (see Section 2.2).

**Pose estimation**

Pose estimation from point correspondences, for calibrated cameras, has been extensively studied in literature. For an intuitive visualization of the geometry of planar pose estimation problem, Fig. 5.1 may be of help. Keypoints $x_i = (u_i, v_i)$, located on the camera imaging sensor (bottom left plane), are in one-to-one correspondence with points $X_i$ standing on a flat reference object (upper right plane). It can be assumed, without loss of generality, that the reference object lays on the $z = 0$ plane of the world coordinate frame, so that all 3D points $X_i$ possess third null coordinate. The set of corresponding 2D-3D points $(x_i, X_i)$, of which $\widetilde{x}, \widetilde{X}$ are just the homogeneous notations, are related by projective equations involving the internal camera matrix K, the rotation matrix R and the translation vector t

$$s\widetilde{x} = A\begin{bmatrix} R & t \end{bmatrix}\widetilde{X} \tag{5.1}$$

Both R and t can be retrieved up to a scalar value $s$ provided that enough corresponding pairs $(x_i, X_i)$ are available and the camera is internally calibrated.

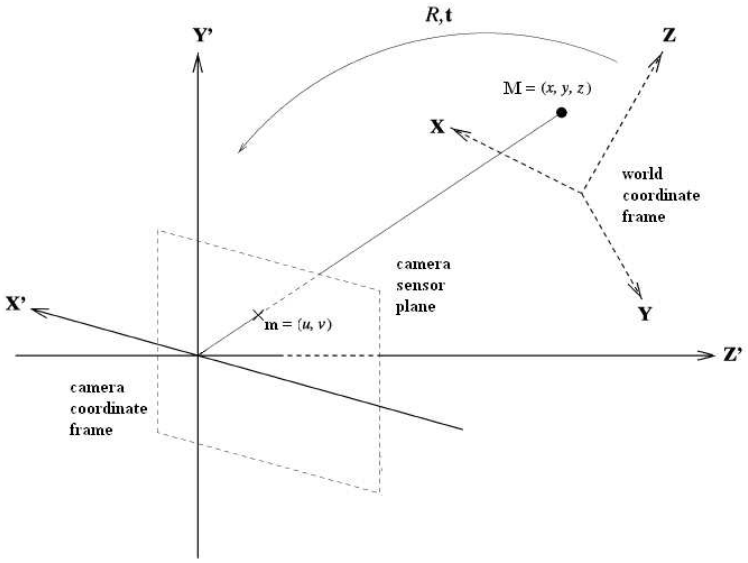



**Figure 5.1:** Geometry of pose estimation from 2D-3D correspondences problem.

Nonetheless, two well known algorithms, addressing the problem from very diverse points of view, have been employed to emphasize flexibility and effectiveness of our

proposal. The first algorithm has been illustrated by Simon et al. [24], and has been considered for long the classical photogrammetric formulation. In practice, they solve for the unknown pose by minimizing the following objective function:

$$\sum_{i}^{N} \| \left( \widehat{u_i} - \frac{R^1 X_i + t_x}{R^3 X_i + t_z} \right), \left( \widehat{v_i} - \frac{R^2 X_i + t_y}{R^3 X_i + t_z} \right) \|^2 \tag{5.2}$$

where $R^i$ is the $i^{th}$ row of matrix $R$ and $t$ is a $3 \times 1$ vector. This computation minimizes the error distance among projections in the image space. In place of the sequential estimation proposed in their paper, we compute the pose of each frame with respect to our appearance model thus avoiding potential estimation drift issues.

Theoretically, an equivalent reformulation of the problem consists in estimating $(R, t)$ that relates the known reference points $X_i$ with the corresponding $X_i^{'}$ so that:

$$X_i^{'} = RX_i + t \tag{5.3}$$

where $X_i = (X_i, Y_i, Z_i)$ and $X_i^{'} = \left(X_i^{'}, Y_i^{'}, Z_i^{'}\right)$ are expressed in an object-centered and camera-centered reference frame respectively. From this viewpoint, the second algorithm, proposed by Schweighofer et al. [22], aims at minimizing an object space error by means of the line-of-sight projection matrix $\widehat{V_i}$. This algorithm yields the best results according to a recent analysis of the state-of-the-art carried out in [18].

Once the pose is retrieved it is then possible to project $3D$ models in the image according to $(R, t)$ and the known camera intrinsics.

### 5.1.3 Experimental results

This section reports the performance of the pose estimation algorithms, presented in Section 5.1.2, in two different case studies. Performance are measured in terms of estimation steadiness and smoothness. Under this perspective, the most stable the estimated pose over time the better the algorithm. In the following we plot the recovered position of the camera center coordinates $O^C = \left(O_X^C, O_Y^C, O_Z^C\right)$ expressed in the object-centered frame. Both algorithms are run twice on each sequence with different appearance models; the first time using a single image (Fig. 5.2 top), the second time using a mosaic (Fig. 5.2 bottom). All the frames used to build the models do not belong to the test sequences.

The two test sequences have been acquired by a freely moving observer using a consumer grade web camera, a Logitech Quick Cam Sphere. Each sequence is about 600 frames long and images have a resolution of $640 \times 480$ pixels.

**Figure 5.2:** Small (top) and large (bottom) appearance models.

### Aeronautical servicing

The first case study is drawn from a collaborative research project called ARIS (Augmented Reality to Increase Safety) that addresses the application of Augmented Reality into the field of aeronautical servicing. The ultimate aim of the project is to equip engineers with see-through goggles by which a context-aware system will act as a virtual assistant providing information on the maintenance procedure in real-time using augmented reality. The sequence portraits the inside of a cockpit of a plane. Useful information in this context concerns the position of the most important switches and leverages as well as instructions on how to operate them properly (refer to Fig.5.4 for some examples).

In the upper row of Fig.5.3 the position of $O^C$ according to the pose estimated using a small appearance model is reported. While the pose is correct most of the time, the peaks in the plots denote that the estimation suffers from jitter. Notably, both pose estimation methods are affected by these peaks approximately in the same way. Conversely, the plots in the lower row of Fig.5.3 show that, when using the mosaic as appearance model, the estimated pose exhibits a much smoother trend and jitter is almost completely eliminated, with the exception of some creases on the $z$ component. It is also worth noticing the proposed approach yields accurate and convincing video augmentation also in presence of significant image brightness changes, as shown by Fig.5.4.

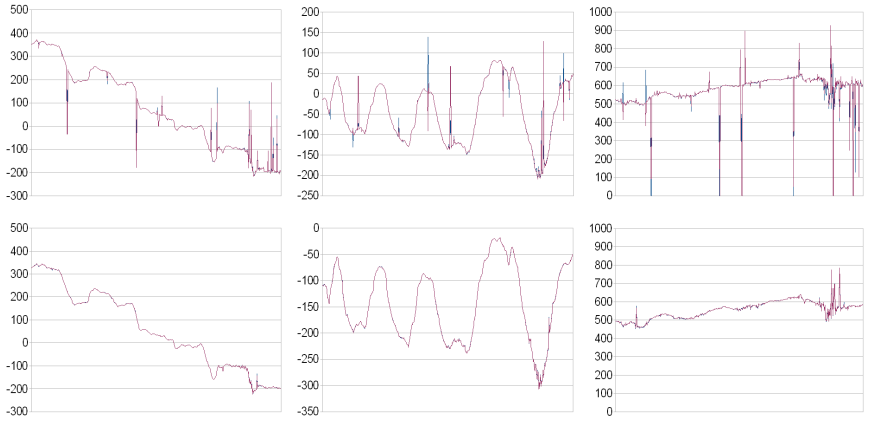



**Figure 5.3:** Recovered camera center coordinates using small (top) and large (bottom) appearance models: Schweighofer et al. (violet), Simon et al. (blue). Left to right: $O_X^C$, $O_Y^C$, $O_Z^C$.

**Figure 5.4:** Augmented cockpit sequence samples.

### Cultural heritage

The second case study concerns an advanced context-aware system for delivering information to visitors of museums or archaeological sites, by means of Augmented Reality. The considered sequence has been acquired at the Archaeological Museum in Bologna and displays a showcase with Etruscan jewellery. Fig. 5.6 shows that the pose of the observer with respect to the showcase is accurately retrieved, as vouched by the coloured outlines superimposed on the borders of the shelves. Besides, additional context aware information is conveyed by highlighting the object that is likely to be the most important for the user given his position and orientation.

As before, the estimation using a small appearance model is quite good but suffers from jitter (as it can be seen in the upper row of Fig.5.5). When using the mosaic (lower row of Fig.5.5), jitter mostly disappears and, unlike previous experiment, the pose is smoother even when there are no macroscopic estimation error.

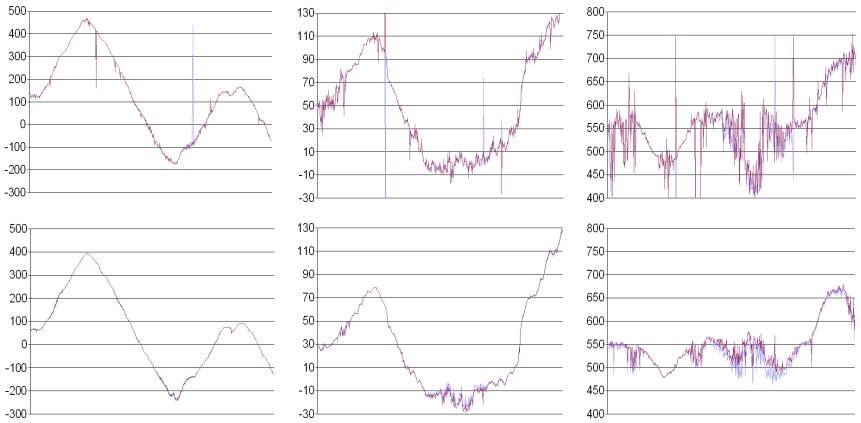



**Figure 5.5:** Recovered camera center coordinates using small (top) and large (bottom) appearance models: Schweighofer et al. (violet), Simon et al. (blue). Left to right: $O_X^C$, $O_Y^C$, $O_Z^C$.

**Figure 5.6:** Augmented samples from jewellery sequence.

### 5.1.4 Conclusions

In this section we have described an approach to augmented reality that is suitable to environments where large planar objects are present. Instead of modeling the reference objects using a single image or a set of independent images, we propose to build a mosaic by registering together several detailed views. The pose is then estimated from the correspondences between the actual frame and the appearance model of the reference planar object using known pose estimation algorithms. Experiments demonstrate that two very different pose estimation algorithms largely benefit from the proposed approach. In this sense our proposal can be thought as a preprocessing step able to improve the computational performance and accuracy of any pose estimation algorithms.

## 5.2 Vision-based markerless gaming interface

This section discusses a novel human machine interface for gaming applications based on computer vision. The key idea is to allow the user to interact with the game by simply moving a hand-held consumer grade camera. Detection of natural features in the incoming video stream avoids instrumenting the scene with optical markers while preserving real-time computation and accuracy. A prototype videogame developed as proof-of-concept of the camera-based gaming interface is also presented. Thanks to recent advances in real-time extraction and matching of natural features from images on mobile platforms, our proposal holds the potential to enable a new generation of camera-controlled videogames for hand-held mobile devices.

### 5.2.1 Introduction

The ever increasing pervasiveness of computer systems into our everyday environment calls for novel mechanisms of human-computer interaction. Interfaces to computerized equipment need to be straightforward and effective, the ability to interact using inexpensive tools being highly regarded.

In the last decades, keyboard and mouse have become the main interfaces for transferring information and commands to computerized equipment. In some applications involving 3D information, such as visualization, computer games and control of robots, other interfaces based on remote controller [19], joysticks and wands can improve the communication capabilities despite being sometimes impractical or limited.

Wearable and handheld devices, such as datagloves, “backpacks” [3] and haptics, are designed to be more user friendly, helping untrained users in performing complex tasks. On the other hand, the high cost and cumbersome hardware limit the field of usability of these solutions.

In daily life, however, vision and hearing are the main channels through which humans gather information about their surroundings. Therefore, the design of new interfaces that allow computerized equipment to communicate with humans by understanding visual and auditive input may conjugate effectiveness, naturalness and affordable prices.

Vision based interfaces hold the potential to communicate with computerized equipment at a distance and the machine can be taught to recognize and react to human-like feedbacks. Despite many advances have been recently reached in the field of human gesture, motion and behavior understanding [11, 26, 12], engineers have been mostly focusing on marker-based tracking systems for vision-based human-computer interaction applications. The gaming industry is recently showing a growing interest for vision based interfaces, with many proof of concepts developed so far [8, 27, 2, 20].

As a matter of fact, visual markers can be reliably tracked [6] at low computational costs, although game boards/controllers must be instrumented with them.

Conversely, our proposal deals with a novel vision-based gaming interface able to deliver position and orientation of the player by simply using a hand-held consumer grade camera and without requiring any visual marker. The proposed approach is straightforward since the movement of the camera directly translates into 3D commands to the game and requires no instrumentation of the environment. It is also very effective since camera pose is estimated with millimetric precision. Finally, it is cheap since it relies on widely available low-cost cameras.

### 5.2.2 Related work

Recent works in literature show that, to some extent, human behaviour understanding using imaging devices is attainable. Harville and al. [11] conceived a robust algorithm for 3D person tracking and activity recognition. The work by Viola and Jones [26] paved the way for sound automatic face detection. Isard and al. [12] demonstrated reliable tracking of deformable objects in presence of occlusion and cluttered environments. These outstanding achievements have inspired the work of Lu [17, 16] on vision-based game interfaces controlled respectively by head and hands movements. Head, face and body position tracking for computer games was also successfully demonstrated in the work of Freeman et al. [7]. However, despite being very flexible and natural interfaces from a human perspective, the underlying technology is still computational too intensive to guarantee short latency time and smooth operations. Moreover precise handling and maneuvering tasks demand a detection and reconstruction accuracy that, in some cases, current algorithms may not deliver.

Tracking of optical markers has rapidly emerged as a fast and accurate alternative for conveying simplified information to computer systems. Although complex human behaviours cannot be captured, location and orientation information can be robustly retrieved in a wide variety of environmental conditions and at low computational cost. Examples of videogames built on top of optical marker trackers have been growing steadily in recent years. Cho et al. [2] described an augmented reality shoot-em-up game in which players aim at virtual opponents rendered on a game board filled with optical markers. Oda et al. [20] developed a racing game where users steer their virtual cars using controllers stuck with markers monitored with cameras. Govil and al. [8] designed a marker-based golf ball tracker used to set speed and direction of a virtual ball in a golf simulator. By exploiting the implementation of a marker tracker for portable devices, Wagner and colleagues [27] developed an Augmented Reality (AR) game where multiple players are allowed to interact using camera-equipped PDA devices.

Nonetheless, recent advances in the field of object recognition showed that accurate pose estimation and tracking can be achieved without the need of specific visual markers, but instead using keypoints extracted from textured areas [15]. In particular, the SURF (Speeded Up Robust Features) algorithm [1] reconciled accuracy and low computational cost for robust keypoints extraction and tracking.

Therefore we propose to deploy a camera pose estimation approach based on natural keypoints correspondences as a novel human-machine interface for gaming purposes. It is worth pointing out that camera pose estimation using natural keypoints on mobile phones has been recently demonstrated by Wagner et al. [28]. Hence, our proposal holds the potential for development of new camera-controlled gaming applications for hand-held mobile devices such as phones and PDAs. The remainder of the section describes the camera pose estimation algorithm in terms of its key components and present a prototype videogame, dubbed Black Hole, developed so far as proof-of-concept of our proposed approach.

### 5.2.3 Markerless pose estimation

The interface consists essentially of an automatic camera pose estimation algorithm for scenes in which flat objects are present, therefore limiting the types of suitable scenes. In this case, however, the limitation is slight, since the requirements is that a plane be visible, even if partially occluded, in the scene. This is common in indoor environments, where a textured ceiling or ground plane is usually visible. Outdoors, even rough ground (grass, roads or pavements), provide also an acceptable reference for the system.

The pose recovery algorithm is largely inspired by the camera tracker illustrated by Simon et al. [24], for it delivers accurate estimation at low computational cost. However, differently from the original formulation, pose recovery is performed every time with respect to a reference frame (pose detection) instead of arising from the composition of multiple pairwise registration (pose tracking) among subsequent frames. Hence, pose detection tolerates failures since each frame is processed independently; besides it does not suffer from the dead reckoning issue typical of pairwise composition. On the other hand, pose detection requires a reference object to be known beforehand, i.e. the object with respect to which the pose is continuously computed. Moreover, pose jittering may arise since temporal correlation is usually not reinforced. In the rest of this section the solutions to these two problems are addressed and described.

Using natural keypoints instead of markers makes the instrumentation of the scene not needed anymore since any flat object can be a suitable reference. Just before starting a gaming session a brand new natural reference is learnt on-the-fly by simply taking a snapshot of a textured planar object and extracting a vector of keypoints descriptors.

The corresponding points of the reference keypoint set are searched within every new incoming frame and pairs of matching keypoints are likely to be detected even in case of large pose and illuminations changes, as shown in Fig. 5.7. Incorrect keypoints pairs can be easily detected and discarded using a RANSAC-based homography estimation step [10]. The remaining corresponding pairs are linked by the geometric relationships explained in Section 5.1.2, hence they are fed to a pose estimation algorithm, for example the one described in [24], in order to obtain a reliable estimation of the position and orientation of the camera with respect to the reference object. Differently from the mosaic-based approach described in Section 5.1, here pose estimation relies on a single-image description of the reference object. This choice is tightly connected with the intended application; since gaming interfaces have to be as simple and practical as possible, acquisition of a single snapshot is quicker and easier than that of multiple views or a video.

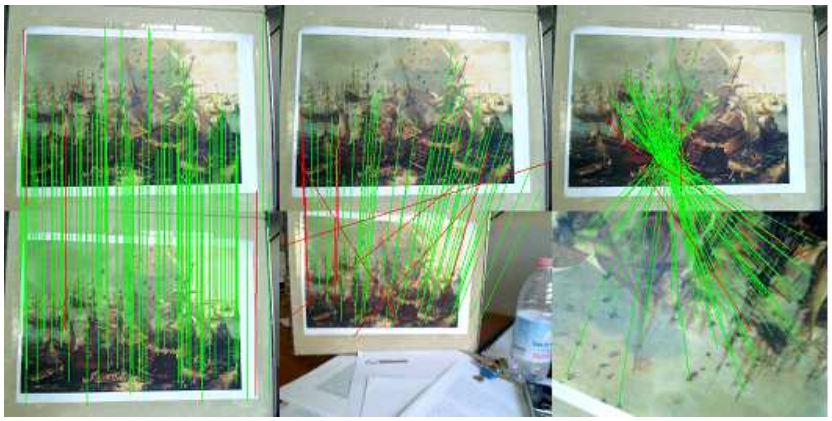


**Figure 5.7:** Tracking SURF keypoints in few snapshots taken from different viewpoints: correct (green) and incorrect (red) corresponding pairs.

Nonetheless, delivered poses still exhibit an excellent accuracy with camera position usually estimated in the range of few millimeters from the true one. Nonetheless, since this approach does not exploit the temporal continuity of the camera trajectory, the sequence of estimated poses usually exhibit jitter effects. This problem manifests as small vibrations among subsequent estimations, such discontinuities being quite noticeable by a human observer and tending to degrade the gaming experience. In order to mitigate this effect a pose smoothing technique has been adopted. The adopted approach, described in [21], consists in linking every new pose with those computed during a previous time window by exploiting a Support Vector Regression scheme as a temporal regularization term.

Natural keypoint correspondences and pose smoothing make the conceived pose

estimation algorithm fast, robust and practical, thus providing accurate and jitter-free estimations without the need for fiducial markers placed all across the scene.

### 5.2.4 Gaming application

A prototype videogames has been developed using as interface the vision-based pose estimation algorithm described previously. In addition, few third-party libraries have been integrated for a number of specialized tasks, in particular:

- OpenGL (Open Graphics Library) [9], a portable and interactive 2D and 3D graphics library adopted for fast visualization and rendering.
- OpenCV (Open Computer Vision) [4], a collection of computer vision functions used for video capturing, keypoints detection and numerical optimization.
- Tokamak [13], an open-source real-time physics engine used for accurate simulation of dynamics of rigid body, gravity, friction and so on.

The typical hardware configuration used to run the games consists of a single laptop PC powered by an Intel Core 2 CPU, equipped with 4 GB RAM and running Windows XP. The video camera is a Logitech Quick Cam Sphere grabbing color sequences at $640 \times 480$ resolution. The game has been developed in C++ using Microsoft Visual Studio 2005. Using this setting the frame rate ranges between 6 and 10 frames per second (FPS), keypoints extraction being the major bottleneck of the system. Although quite far from real-time processing, the system is responsive enough to allow for a satisfactory gaming experience. By reducing the camera resolution to $320 \times 240$ the frame rate increase to 9 - 15 FPS without severely penalizing accuracy.

#### Black Hole

Black Hole is a puzzle game inspired by the dark atmosphere of Star Wars. The goal is to steer a R2D2-like ball through a Death Star maze till the endpoint avoiding the holes spread along the path. The user can slant and rotate the maze by moving a webcam held in his hand. Gravity effect allows the user to control the ball by moving the maze; friction and collision against maze walls and floor are also implemented in order to add realism. Every time the user loses a ball, by letting it fall in a hole, it obtains a number of points commensurate to the distance from the starting point. After three lost balls the game ends and the final score is the sum of the points obtained thus far.

Figure 5.2.4 shows the starting and ending screens of Black Hole together with some screenshots taken during a gaming session. Figure 5.8 shows some images taken by the webcam hand-held by the player and the corresponding game screen, the reference object being a textured picture printed on a paper sheet and laying on the desktop.


Black Hole starting screen (left) and game ending (right).

The image pairs, screenshot and camera frame, show how the floating maze is tilted according to the instantaneous orientation of the hand-held camera with respect to the reference object.

**Feedback and observations**

The game has been on show for few weeks in our laboratory rooms and has been played by some colleagues from other labs that gently provided feedbacks and suggestions. First of all, only a picture of a person pointing the camera to the reference pattern laying on the table has been required by anybody to start playing the games. Such a limited amount of training information hints at the ease of use and naturalness of the conceived interface. Most of the players manage to get to the end of the game, this suggesting also good intuitiveness and friendliness. On the other hand several persons expressed concerns about the difficulty of keeping the reference object always in sight during the gaming session. Even though occasional pose estimation failure does not necessarily ruin the game experience, it might be annoying especially during fast and critical phases. Another set of complains concerns the responsiveness of the gameplay which is mainly accountable to the high computational cost that the system incur when highly textured areas generate a large amount of keypoints.

### 5.2.5 Conclusions and future work

The ubiquitous presence of computerized equipments in everyday environment calls for conception and design of natural and easy-to-use human-machine interfaces. Practical, straightforward and inexpensive are the keywords for the next generation of interaction paradigms. Videogames are a challenging test ground since fast response and high accuracy are also required. Vision-based interfaces hold the potential to fulfill this expectations. A vision-based approach based on tracking natural features has been

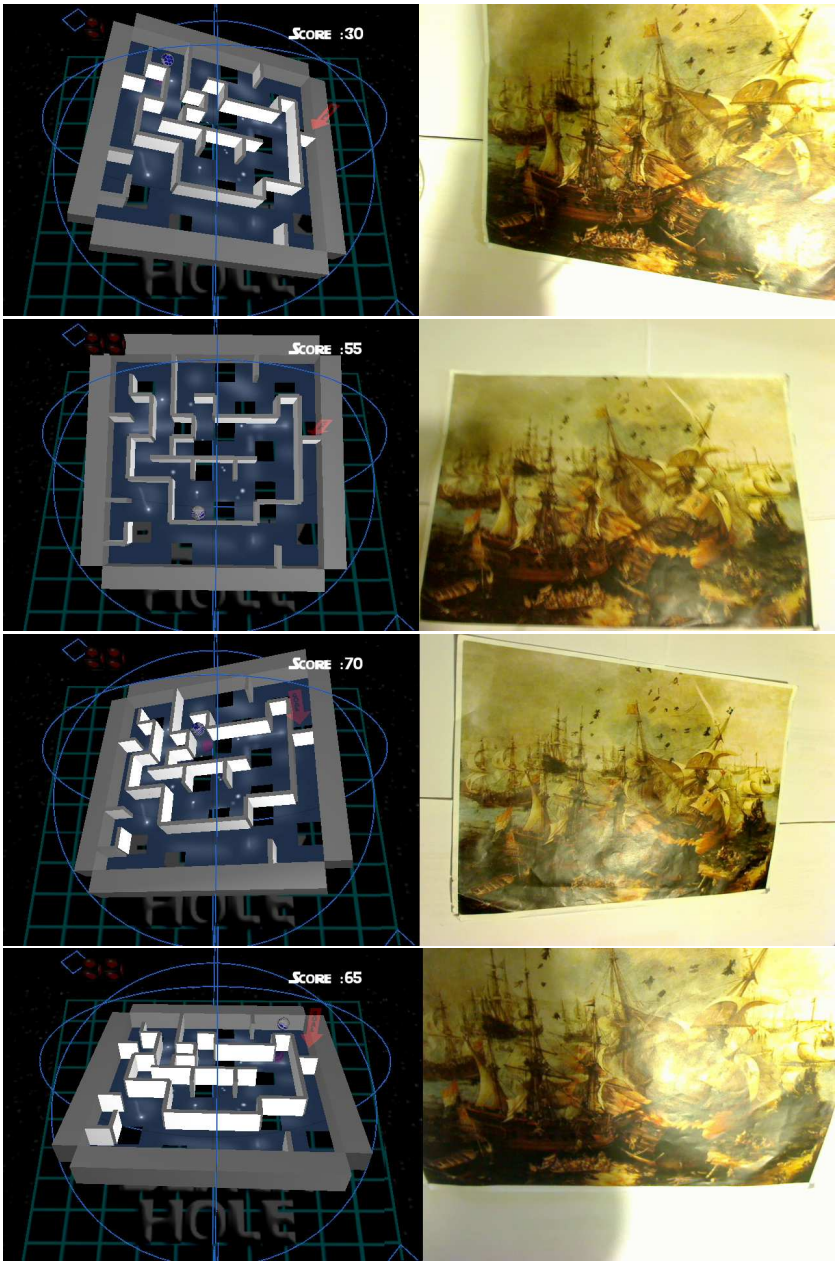



**Figure 5.8:** In the left column, 4 snapshots depict the maze, tilted in different ways, according to the orientation of the camera with respect to the reference object computed in each frame (right column).

conceived as an interface for gaming applications. The proposed approach allows the user to interact with a videogame by simply moving a webcam pointing toward a planar

textured object present in the scene. According to the feedback received by several users, the interface is intuitive, fast, responsive and, ultimately, enjoyable.

As for future directions of works, pose estimation from non-flat surfaces or larger-than-a-single frame object would prove useful to increase the possibility for the user to move around. Moreover, as for the difficulty of keeping the reference object always in sight, we wish to investigate on the possibility of enabling also a mixed-reality mode, in which the user would see the virtual objects of the game superimposed to actual video stream coming from the camera.

Eventually, the proposed approach is particularly suited to enable gaming applications on hand held devices such as phones and PDA, for the user may simply point the integrated camera toward a textured plane and play by moving the device in his hand. Therefore, in the near feature we plan to port our gamimg interface on a state-of-the-art hand held device.

# Bibliography


[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[2] K. Cho, W. Kang, J. Soh, J. Lee, and H. S. Yang. Ghost hunter: a handheld augmented reality game system with dynamic environment. In *Proc. of Intl. Conf. on Entertainment Computing*, pages 10–15, 2007.

[3] B. Close, J. Donoghue, J. Squires, P. De Bondi, M. Morris, W. Piekarski, and B. Thomas. Arquake: an outdoor/indoor augmented reality first person application. In *Proc. of IEEE Intl. Symp. on Wearable Computers*, pages 139–146, 2000.

[4] Intel©Corp. Opencv 1.0, open source computer vision library, 2000-2007.

[5] C. Dehais, M. Douze, G. Morin, and V. Charvillat. Augmented reality through real-time tracking of video sequences using a panoramic view. In *Proc. of Intl. Conf. on Pattern Recognition*, pages 995–998, 2004.

[6] M. Fiala. Artag, a fiducial marker system using digital techniques. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 590–596, 2005.

[7] W. T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma. Computer vision for computer games. In *Proc. of Intl. Conf. on Automatic Face and Gesture Recognition*, pages 100–105, 1996.

[8] A. Govil, S. You, and U. Neumann. A video-based augmented reality golf simulator. In *Proc. of ACM Multimedia*, pages 489–490, 2000.

[9] Khronos Group. Opengl 2.1, open computer graphics library, 1992-2008. `http://www.opengl.org/`.

[10] R. Hartley and A. Zisserman. *Multiple view Geometry in computer vision*. Cambridge University Press, Second Edition, 2003.

[11] M. Harville and D. Li. Fast, integrated person tracking and activity recognition with plan-view templates from a single stereo camera. In *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition*, pages 398–405, 2004.

[12] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Intl. Journal of Computer Vision*, 29(1):5–28, 1998.

[13] D. Lam. Tokamak, open physics engine library. `http://www.tokamakphysics.com/`.

[14] P. Liu, X. Sun, N. D. Georganas, and E. Dubois. Augmented reality: a novel approach for navigating in panorama-based virtual. In *Proc. Intl. Workshop on Haptic, Audio and Visual Environments and their Applications*, pages 13–18, 2003.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60(2):91–110, 2004.

[16] P. Lu, Y. Chen, X. Zeng, and Y. Wang. A vision-based game control method. In *Proc. of Intl. Conf. on Computer Vision, Workshop on Human Machine Interaction*, pages 70–78, 2005.

[17] P. Lu, X. Y. Zeng, X. Huang, and Y. Wang. Navigation in 3d game by markov model based head pose estimating. In *Proc. of Intl. Conf. on Image and Graphics*, pages 493–496, 2004.

[18] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative o(n) solution to the pnp problem. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 1–8, October 2007.

[19] Nintendo©. Wii. `http://wii.nintendo.com/`.

[20] O. Oda, L. J. Lister, S. White, and S. Feiner. Developing an augmented reality racing game. In *Proc. of Intl. Conf. on Intelligent Technologies for Interactive Environment*, 2008.

[21] S. Salti and L. Di Stefano. Svr-based jitter reduction for markerless augmented reality. In *Proc. of Intl. Conf. on Image Analysis and Processing (submitted paper)*, 2008.

[22] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(12):2024–2030, 2006.

[23] G. Simon and M. Berger. Real time registration of known or recovered multi-planar structures: application to ar. In *Proc. of IEEE British Machine Vision Conference*, pages 567–576, 2002.

[24] G. Simon, A. W. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proc. of Intl. Symposium on Augmented Reality*, pages 120–128, May-June 2000.

[25] Y. Uematsu and H. Saito. Vision-based registration for augmented reality with integration of arbitrary multiple planes. In *Proc. of Intl. Conf. on Image Analysis and Processing*, pages 155–162, 2005.

[26] P. Viola and M. J. Jones. Robust real-time face detection. *Intl. Journal of Computer Vision*, 57(2):137–154, 2004.

[27] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg. Towards massively multi-user augmented reality on handheld devices. In *Proc. of Intl. Conf. on Pervasive Computing*, pages 208–219, 2005.

[28] D. Wagner, G. Reitmayr, Alessandro Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In *Proc. of Intl. Symp. on Mixed and Augmented Reality*, pages 125–134, 2008.

# Chapter 6

# 3D reconstruction of deformable surfaces

This chapter investigates sparse geometric reconstruction of objects using a set of images. Differently from conventional structure from motion algorithms that usually deal with rigid objects, an innovative method for fast shape retrieval of deformable objects relying on a single camera is detailed.

The shape reconstruction problem is tackled by describing a deformable object with a tesselated surface, for instance a triangulated mesh, with a sufficient level of detail, i.e. number of triangles. Assuming the region of object inside a triangle as being flat, the geometric reconstruction of the whole mesh amounts at computing the homography between each triangle of the model and its corresponding projection in a given image. The extension to deformable objects requires to properly constrains each homography considering that every triangle is connected to others inside the mesh, and any solution must maintain continuity across the mesh. Moreover, smoothing constraints must be included to prevent unrealistic deformations to produce high likelyhood estimates.

A re-parametrization of the problem in terms of the vertex coordinates of the triangulated model has been envisioned, thus permitting to specify continuity and smoothing constraints in an elegant and concise formulation. The devised framework admits also a fast iterative linear solver, based on projection kernels, boosting the computation performance of the algorithm. The algorithm recovers the shape of a deformable surface using 3D-2D correspondences computed from natural texture, thus not requiring any instrumentation of the scene.

Thanks to a ongoing collaboration between the Ecole Polytechnique Federal of Lausanne and Solar Impulse SA [6], the conceived approach has been tested in a challenging real scenario. Solar Impulse is an ambitious project aimed at realizing the first,

solar propelled, airplane able to trip around the world without exploiting fossil energy. Since the wings of SolarImpulse will be both very long and very light, they must be monitored accurately both for safety and efficiency. The proposed algorithm has been deployedfor for measuring wing deformations of the SolarImpulse scaled model prototype. Performance assessment using both synthetic and real data is reported in the last section of the chapter.

## 6.1 Shape recovery of non-rigid objects

Experimental determination and measurement of wing deformations is of fundamental importance for the analysis of structural dynamics in the aerospace industry. Knowing the way wings deform during flight could provide valuable information for testing the validity of finite elements analysis and for improving the design and manufacturing process.

Present methods of measuring wing deformations usually entail the instrumentation of the aircraft, i.e. a set of accelerometers or strain gauges placed all over the aircraft. Despite being accurate, such methods are invasive and might influence the dynamics and, eventually, the measurements (i.e. added mass due to instrumentation). Moreover, these sensors can only measure deformations, along a single direction, at a few preset locations and are difficult to move once the wing is constructed.

Since vision-based approaches are known to provide dense measurements through non-contact sensing, some works based on imaging devices have been attempted. The work by Ryall and al. [14] shows how three dimensional modes of an oscillating wing section can be recovered by tracking visual markers stuck on it. However it requires special hardware, i.e. synchronized strobe lights and camera, and performs off-line. Recently, Barrows [3] has proposed a multiple-camera system for on-line reconstruction of a wing inside a wind tunnel. Both the approaches require cumbersome hardware and the instrumentation of the aircraft, making them expensive and impractical for the acquisition of measurements during the flight.

This section describes a vision-based on-line approach for measuring wing deformations that relies on a single camera and on “natural markers”, i.e. textured areas underneath the wings. By requiring just a single camera, this method is a cheap and practical way of evaluating the behavior of wings in real conditions.

To validate this technique and demonstrate that it can be deployed in a realistic aeronautical context, a complete pipeline designed to measure the deformations of SolarImpulse’s [6] scaled model wings has been put in place. This is an interesting test case because the wings of SolarImpulse will be both very long and very light. As a result, they are bound to deform noticeably in flight and it will be important to verify that

they behave as expected. Experiments shows that measurements accuracy up to few millimeters can be achieved monitoring a 4-meters wide model of the Solar Impulse with a consumer grade camera.

### 6.1.1 Related work

Monocular 3D shape recovery of deformable surfaces is known to be an ill-posed problem even when there is sufficient texture for structure-from-motion and template-matching approaches to be effective. A priori knowledge of deformation models is required to solve ambiguities and make the problem tractable.

Structure-from-motion methods rely on feature points tracked through a sequence to retrieve the deformed shape of a surface [9, 15]. However, the underlying linearity assumptions of these methods limit their applicability to smooth deformations. The use of more generally applicable constraints have been advocated [18, 16], even though additional assumptions, that may not apply, are required.

Statistical learning approaches have therefore become an attractive alternative that takes advantage of observed training data. Linear approaches have been applied to faces [4, 10] as well as to general non-rigid surfaces [16]. However, they impose the same restrictive smoothness constraints as before. Moreover, training the model of highly deformable surfaces represented by meshes with many vertices, and therefore many degrees of freedom, requires a number of training examples that quickly becomes intractable.

Another class of approaches solve this problem by introducing a physical model that can infer the shape of untextured surface portions from the rest of the surface [12, 11]. Due to the high dimensionality of such representations, modal analysis [15] was proposed to model the deformations as linear combinations of modes. Some knowledge about the surface material must be assumed since the deformation model is defined in terms of physical parameters. Moreover the complexity and non-linearity of the true physics make physically-based approaches an accurate approximation only in case of small deformations.

Since one can reasonably assume that aircraft wings are made of material whose mechanical property can be known and expected deformations are meant to be small, physical models become a suitable choice in this context. Moreover, a similar approach [15] has been integrated into a software package designed to model the deformations of sails from video sequences and to measure visually their curvature. Delivered to Team Alinghi, it supports the design team by monitoring the behavior of the spinnaker under real sailing conditions, providing valuable informations to improve its design.

Since sails act very much like wings, both may be treated as smooth deformable surfaces and the approach we propose for measuring wing deformations is largely in-

spired by the works of Salzmann and Pilet [16, 15].

## 6.2 Deformable shape recovery

We represent a surface as a 3D triangulated mesh M=$(V, F)$, where V=$(v_1, ..., v_{N_V})$ is the vector of vertices and F is the list of facets. The position of a vertex $v_i$ is specified by its 3D coordinates $(x_i, y_i, z_i)$. The overall shape is therefore controlled by a state vector $S$, that is the vector of all $x$, $y$ and $z$ coordinates. We assume we are given a set of 3D to 2D correspondences between surface points and image locations.

We assume that a mesh deforms to minimize the objective function

$$\epsilon(S) = \lambda_D \epsilon_D(S) + \epsilon_C(S) \tag{6.1}$$

where $\epsilon_C$ is a data term that takes point correspondences into account, $\epsilon_D$ is a smoothness term that tends to preserve the regularity of the mesh, and $\lambda_D$ is a constant.

### 6.2.1 Data term

In this section, we formulate the computation of the 3D mesh vertex coordinates given the data term in terms of solving a linear system. To this purpose we express all world coordinates in the camera referential for simplicity and without loss of generality. Let $X_i$ be a 3D point whose coordinates are expressed in the camera referential. Since we use a single camera and assume its internal parameters to be known, we write its perspective projection as:

$$\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \frac{1}{k_i} \mathbf{A} \left[ \mathbf{I}_{3\times 3} \mid \mathbf{0} \right] \begin{pmatrix} x_i \\ 1 \end{pmatrix} \tag{6.2}$$

where $A$ is the internal parameter matrix and $k_i$ a scale factor. If $X_i$ lies on the facet of a triangulated mesh, it can be conveniently expressed as a weighted sum of the facet vertices, so that (6.2) can be rewritten as

$$\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \frac{1}{k_i} \mathbf{A} \left( a_i \mathbf{v_{i,1}} + b_i \mathbf{v_{i,2}} + c_i \mathbf{v_{i,3}} \right) \tag{6.3}$$

where $v_{i,1\leq i\leq 3}$ are the 3-D coordinate vectors of the vertices and $(a_i, b_i, c_i)$ the barycentric coordinates of $X_i$.

Let's assume that we are given a list of $n$ such 3-D to 2-D correspondences for points lying inside the mesh facets. As pointed out by [16], the $v_{i,1\leq i\leq 3}$ coordinates of

the vertices can be computed by solving

$$\begin{pmatrix} a_1\mathbf{T_1} & b_1\mathbf{T_1} & c_1\mathbf{T_1} & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & a_j\mathbf{T_j} & b_j\mathbf{T_j} & c_j\mathbf{T_j} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_l\mathbf{T_l} & 0 & b_l\mathbf{T_l} & 0 & c_l\mathbf{T_l} & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \begin{pmatrix} \mathbf{v_1} \\ \dots \\ \mathbf{v_{n_v}} \end{pmatrix} = \mathbf{0} \tag{6.4}$$

with

$$\mathbf{T_j} = \mathbf{A_{2\times 3}} - \begin{pmatrix} u_j\mathbf{A_3} \\ v_j\mathbf{A_3} \end{pmatrix}$$

where $\mathbf{A_3}$ represents the last row of matrix $A$ and $a_{2\times 3}$ its first two rows.

### 6.2.2 Smoothness term

Previous work by Salzmann et al. [16] demonstrates that keypoint correspondences do not provide enough independent equations for the problem to be solved uniquely. Two kind of smoothness terms have been used to prevent the estimation of unrealistically deformed shapes:

- stiffness matrix, it carries information about the physical properties of the surface material. Physical properties are expressed by coupling the displacements of neighboring vertices of the mesh. A popular algorithm from [8] has been used to generate a system of equations given a triangulated mesh and few additional parameters such as mass and thickness.
- inextensibility constraints, they model a kind of triangulation that can be thought of as a polyhedron made of metal plates whose edges have been replaced by hinges. Length variations of the edges are discouraged through adding penalties to the overall energy function.

### 6.2.3 Optimization strategy

Differently from the approach in [7] we chose to implement inextensibility constraints exactly. Since such constraints are quadratic, they do not fit in a linear formulation [16]. For that reason an iterative optimization has been conceived.

The idea is to minimize

$$||MX|| \text{ subject to } C(X) = 0\ , \tag{6.5}$$

where $X$ is an $n \times 1$ vector and $C(X)$ an $m \times 1$ vector of constraints.

At each iteration, given the current $X$ state, find $dX$ such that

$$\begin{aligned} C(X+dX)=0 \quad &\Rightarrow \quad AdX=-C(X) \ , \\ &\Rightarrow \quad dX=-A^{\dagger}C(X)+(I-A^{\dagger}A)dZ \ , \end{aligned} \tag{6.6}$$

where $A$ is the $m \times n$ Jacobian matrix of $C$, $A^{\dagger}$ its $n \times n$ pseudo-inverse, and $dZ$ an arbitrary $n \times 1$ vector. In general, $m < n$ and $A^{\dagger}$ can be computed as $\lim_{\delta \to 0} A^t(AA^t + \delta I)^{-1}$, which involves inverting an $m \times m$ matrix and exists even if $AA^t$ itself is non invertible.
Let $P = I - A^{\dagger}A$ be the projector onto the kernel of $A$ and let $dX_0 = -A^{\dagger}C(X)$ be the minimum norm solution of Eq. 6.6. We choose $dZ$ by minimizing

$$\|M(X+dX_0+PdZ)\| \ , \tag{6.7}$$

or, equivalently, solving in the least square sense

$$MPdZ=-M(X+dX_0) \ . \tag{6.8}$$

In this setting, matrix $M$ consists of two parts, the first comes from the data term while the second is made of physical relations encoded in the stiffness matrix. The functional C represents the nonlinear inextensibility constraints.

Since the optimization criterion M weights all the data fairly, gross outliers generate large residuals that could bias the solution. To give outliers a milder impact on the solution, we reformulated the original problem in a reweighted least squares fashion:

$$\| \ WMX \ \| \ \text{ subject to } C\,(X)=0 \tag{6.9}$$

where W is a diagonal weighting matrix. The main diagonal of W is the vector L whose coefficients are computed as follows:

$$L_i = -\exp \frac{d_i}{\hat{d}} \tag{6.10}$$

where $d_i = \| \ F_i \ \|$ is the norm of the $i^{th}$ residuals and $F = MX - b$. $\hat{d} = \frac{1}{N}\sum_i d_i$ is the average of the norm of the residuals. In this settings the reweighted least squares solution is given by

$$\| \ WM\,(X+dX_0+PdZ) \ \| \tag{6.11}$$

## 6.3 Detailed approach

The proposed approach entails the accurate calibration of the imaging device, the presence of a 3D model of the object in its rest position and the capability of establishing correspondences between that model and a given image. The whole approach is split in two main stages:

- Offline phase, it consists of camera calibration and generation of the 3D model in its rest, i.e. undeformed, position
- Online phase, it aims at retrieving pose and deformations given the actual image and the rest model

### 6.3.1 Offline phase

A 3D model of an object in its rest position is composed of two parts: a pointwise model consisting a cloud of points $X_i$ that lie on its surface and a geometric model in the form of a triangulated mesh that approximates its hull. The pointwise model is necessary to detect and establish correspondences between the object and a given image. The geometric model represents a piecewise planar approximation of the true object shape and embeds also the concept of joints, i.e. lines along which the shape is allowed to deform. For the algorithm to perform consistently, the two models have to be spatially aligned, registered pointwise and geometric models will be called hereinafter, just, 3D model of the object.

The construction of the pointwise model is performed using Australis [13], a structure-from-motion software. Given multiple pictures of the same subject taken from different viewpoints, the algorithm is able to generate a sparse cloud of 3D points that reproject consistently in all the views. In detail, the standard reconstruction process take place as follow:

- a set of retroreflective markers, manually placed all over the scene, are automatically detected in every pictures and correspondences are established based on appearance and geometric constraints
- a reference object, shipped together with the software, is used to retrieve an initial estimate of the pose for every single pictures.
- a bundle adjustment solver [2, 5] performs a non linear minimization of the reprojection error across the whole set of correspondences yielding accurate poses and structure

However, the standard process suffered from many drawbacks and a revisioned procedure has been devised to improve reliability and flexibility.

Detailed insights concerning the characterization of the performance of the standard approach and the improvements obtained using the modified method are reported in [1].

Finally, each point $X_i$ belonging to the pointwise model is linked with a vector of all the SIFT descriptors computed in each image in which it appeared. This step is

fundamental to enable subsequent keypoints matching between the pointwise model and a given image.

The geometric model is a triangulated mesh M=$(V, F)$, where V=$(v_1, ..., v_{N_V})$ is the vector of vertices and F is the list of facets, that represents the aircraft in its rest position. The wings have been manually measured and the tesselation roughly follows the joints between distinct parts of the real plane. The topology of the vertices and the facets underwent many changes, the final arrangement may be appreciated in Fig. 6.3, top.

In order to deliver a set of 3D points expressed in terms of barycentric coordinates with respect to the facet of the mesh, pointwise and geometric models must be registered into the same coordinate frame. A useful initial guess is obtained by aligning the eigenvectors of the Principal Component Analysis (PCA) decomposition of the points cloud and the vertices of the mesh. The underlying idea is that the cloud of points is uniformly distributed across the aircraft, an assumption not so far from reality given the symmetry of the texture underneath the wings. Registration accuracy is improved by deploying a subsequent refinement using the algorithm proposed in [17]. This algorithm performs a robust registration of 3D point data to a triangle mesh in presence of outliers and changes in scale. After the registration, points whose distance from the nearest facet is above an acceptance threshold are marked as outliers and removed from the pointwise model. Inliers are converted in barycentric coordinates with respect to the closest facet.

### 6.3.2 Online phase

The scope of the online phase is, given an image $I$, to retrieve pose and deformations of the considered object. The mathematical procedure presented in section 6.2, devised for such goal, requires a 3D model of the object in its rest position and a set of 2D-3D correspondences. The former requirement is fulfilled by performing the steps described in the previous section. The generation of 2D-3D correspondences is accomplished as follows:

- a set of keypoints $x_i$ is extracted from image $I$
- the set of keypoints $x_i$ is matched with the descriptors stored in the pointwise model (see Figure 6.2)
- the matching between keypoints $x_i$ and descriptors of points $X_i$ belonging to the pointwise model naturally defines 2D-3D correspondences relating image projections and 3D points $x_i \leftrightarrow X_i$.

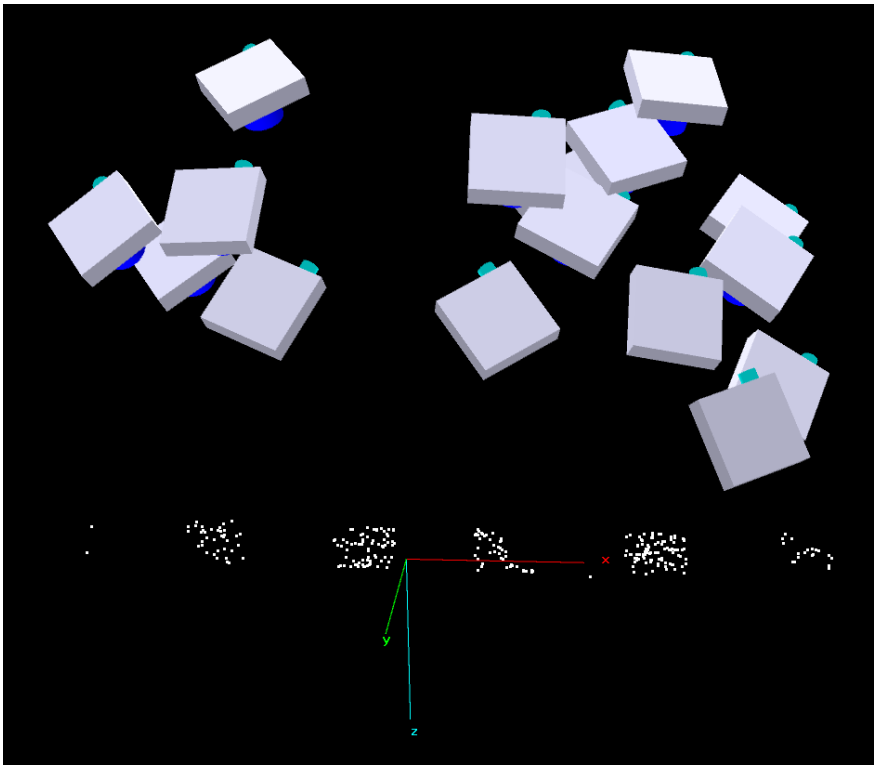



**Figure 6.1:** 3D reconstruction from multiple views. Boxes are cameras, white spots are reconstructed 3d points.

The set of 2D-3D correspondences are then fed to the algorithm. The vertex coordinates of the rest model are the parameters of the state vector $S$ that are to be optimized given the data term, i.e. the 2D-3D correspondences, and the smoothness and continuity constraints. After optimization, the computed state vector $S'$ contains all the coordinates of the vertices and represents the sparse reconstruction of the deformed object observed in image $I$ (see Figure 6.3). Object deformations are defined as the difference between the estimated state vector $S'$ and the vector of coordinates in the rest position $S$.

## 6.4 Results

### 6.4.1 Simulations

Synthetic tests have been conducted to evaluate the performance of the algorithm in a controlled environment and to see how it degrades as different amounts of noise affects the data. The idea has been to create synthetic 2D-3D correspondences using

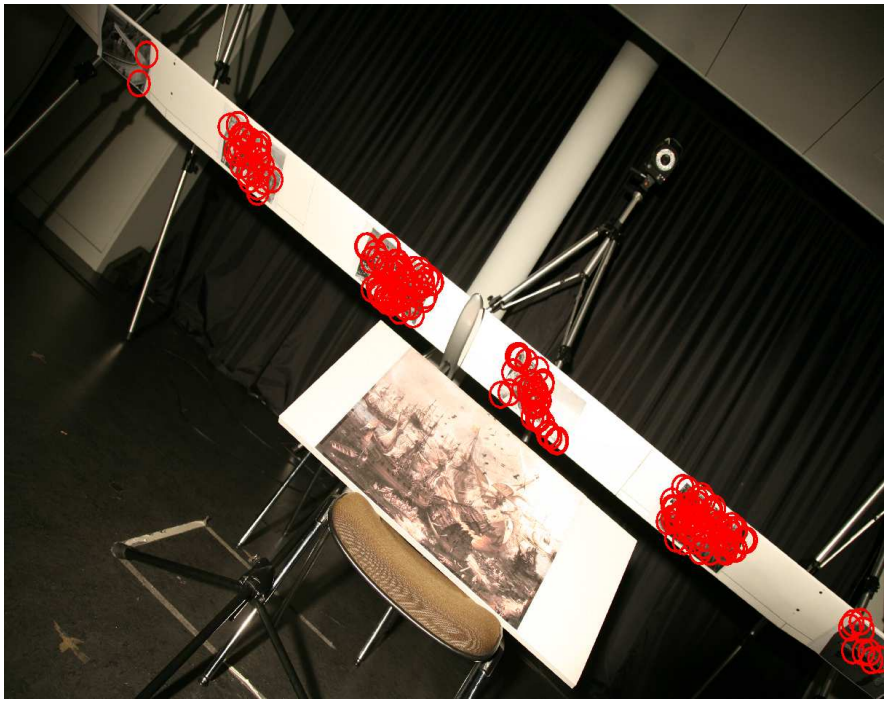


**Figure 6.2:** Keypoints matching. Correspondences between the image and the 3D model are highlighted with red circles.

realistic generative models. As far as 3D points belonging to the pointwise model are conceerned, triangulated mesh describing the aircraft has been deformed designed and deformed by applying a twisting deformation creating a given angle between the two wingtips. Then, a set of 3D points randomly spread over the triangulated mesh have been generated. By projecting the 3D points on virtual cameras randomly spread in the scene, 2D correspondences have been generated. Moreover, uncertainty in the matching process is accounted for by adding gaussian noise to computed projections. The impact of the number and positions of points located on the facets has been analyzed.

The graph in Fig. 6.4 reports on the $Y$ axis the Root Mean Square (RMS) of the difference between the true and the estimated twisting angle, on the $X$ axis the standard deviation of the noise applied to the projections. RMS values have been computed on 1000 trials per number of points.

Remarkably, the algorithm yields high quality reconstructions with realistic amount of noise, i.e. around 1 pixel. Hence accurate wing deformations can be measured using the proposed vision-based approach. Two other facts emerged; the first is the more the correspondences the more precise the reconstruction, the second is that uniformly distributed points are better than scattered ones. Both evidences are quite straightforward,

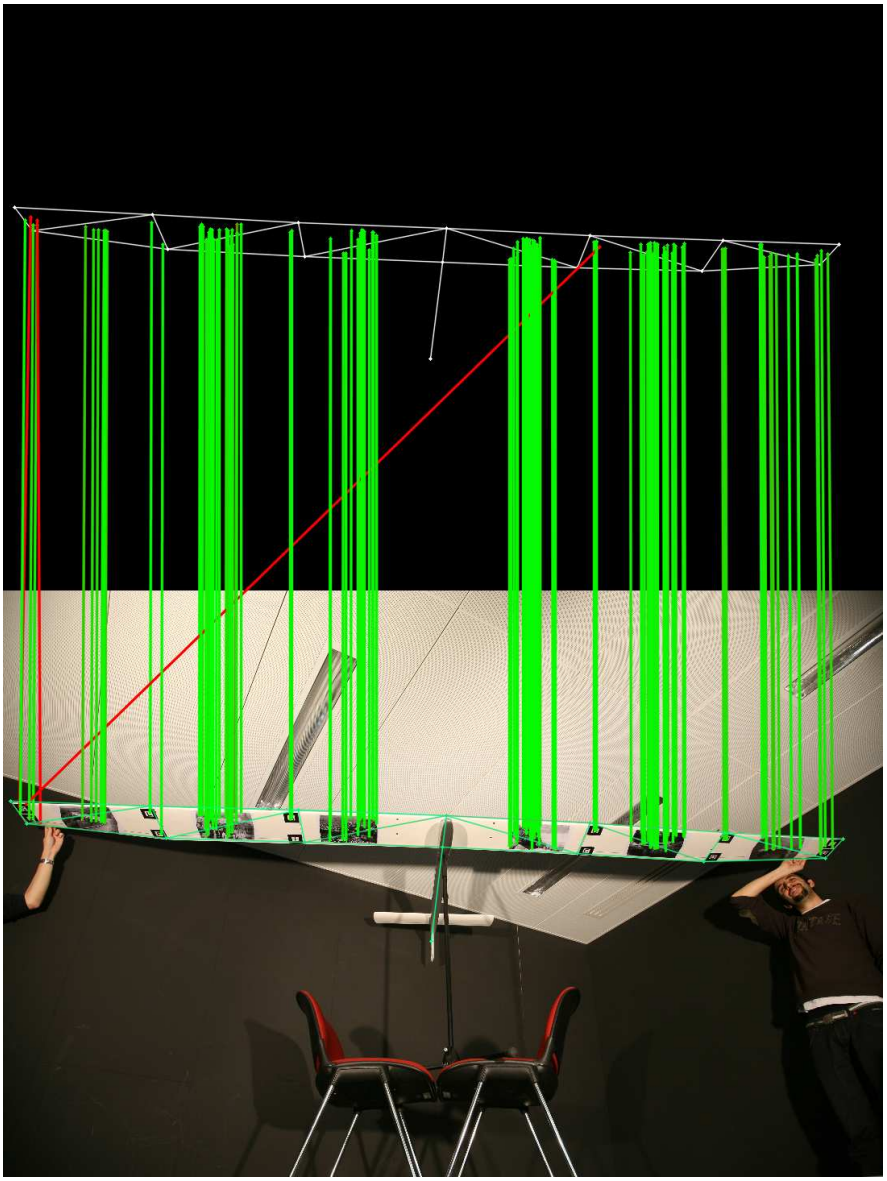


**Figure 6.3:** Deformed geometric model (top), 2D-3D correspondences (coloured lines in the middle), reprojected geometric model on the image (bottom)

perhaps the second remark becomes more interesting when noting that 3 uniform points are better than 5 scattered ones, hence highlighting the importance of the location aside the mere count.

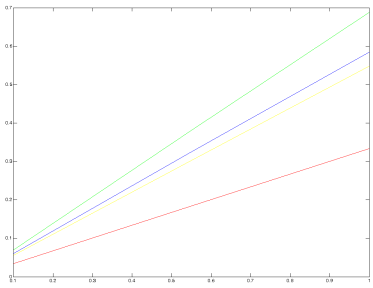


**Figure 6.4:** Twisting angle RMS error with different numbers of points per facet: 3 on uniformly spread (red) and 3 (green), 4 (blue), 5 (yellow) scattered points.

### 6.4.2 Experiments

A wings measurement system for diagnostic and validation purposes requires on-site quantitative performance evaluation. To accomplish this task we decided to obtain ground truth data and compare the attained results to it. But how could ground truth data could be generated?

Two methods, an Optical Motion Capture (OMC) system and a Structured Light Scanner (SLS), accredited of very high accuracy, have been evaluated by assessing the delivered reconstruction in a simple and controllable scenario. The test bed consisted of a stiff beam whose steepness could be carefully set. Since everything was precisely measured, the geometry of any points in the scenario could be carefully computed beforehand and used as ground truth for comparing the methods. With an accuracy of about 0.2 mm, the OMC exhibited the highest level of accuracy, performing an order of magnitude better than the vision-based approach.

Both the OMC and the video-based system have been then deployed in a more relevant setup: measuring wing deformations of the Solar Impulse 4 meter wingspan model. OMC reconstruction has been considered as ground truth data, and the estimation yielded by the video-based approach have been compared with respect to ground truth.

Qualitative and quantitative results are presented. Qualitative results concern the reprojections of the geometric model of the aircraft onto the pictures from which shape has been recovered. As shown in Fig. 6.5 (right), we asked two people to shake the wings during the acquisition to procure deformations. A set of 2D-3D correspondences have been detected using keypoints matching then the optimization procedure jointly determined the deformed model, green mesh (Fig. 6.5), and correct/incorrect matches, drawn respectively with green and red lines. The recovered shape has been reprojected

inside every picture (green mesh) and most of the time it is precisely aligned with the contour of the aircraft (see Fig. 6.6). Fig. 6.5 reports the altitude of the left (red) and right (blue) wingtip computed in each frame of Fig 6.6 with respect to a constant ground plane passing through the body of the aircraft. As can be noted the trends are visually compatible with the deformations (upward then downward) applied by the two persons in the pictures.

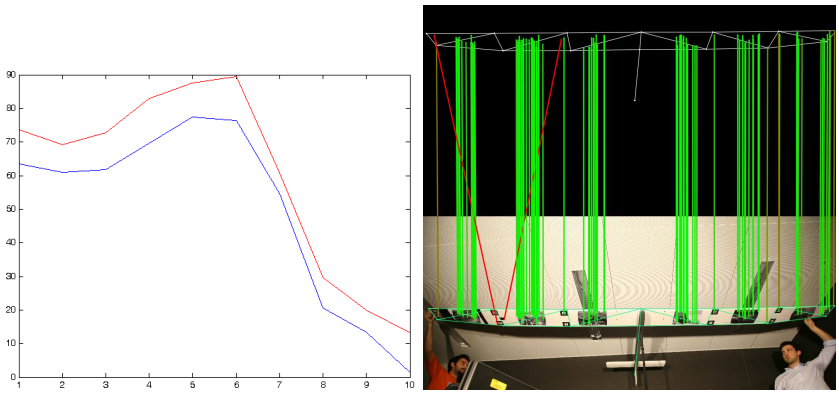



**Figure 6.5:** Wingtips altitude chart (left), estimated mesh reprojection (right).

Video-based 3D reconstruction has been compared with ground truth shape by computing the distance between corresponding points. The RMS error was found to be in the order of 2.5mm. Since the model is 4 meters wide, expectations dictate the measured error to turn to 3 – 4 cm and a 0.5 of twist deviation when coping with the real 60 meters wingspan prototype.

### 6.4.3 Conclusions and future work

The proposed algorithm has shown potential for accurately recovering the shape of large deformable surfaces such as aircraft wings. This is a very important achievement since it may be used for accurate, cheap and non contact measurement of aircraft wings deformations during flight.

A quantitative validation process using the SolarImpulse scaled model attests that an error in the order of 2 mm over a 4 meter wingspan model has been delivered by the system. This error translates on a deviation of about 0.5 degree affecting the twisting angle.

Nevertheless, there is still room for improvements by integrating improved physically-based deformation models, integrating over time and explicitly representing uncertainty in the equations. This is what we will endeavor to do in the future. Furthermore, the use of additional cameras should provide a further increase of accuracy.

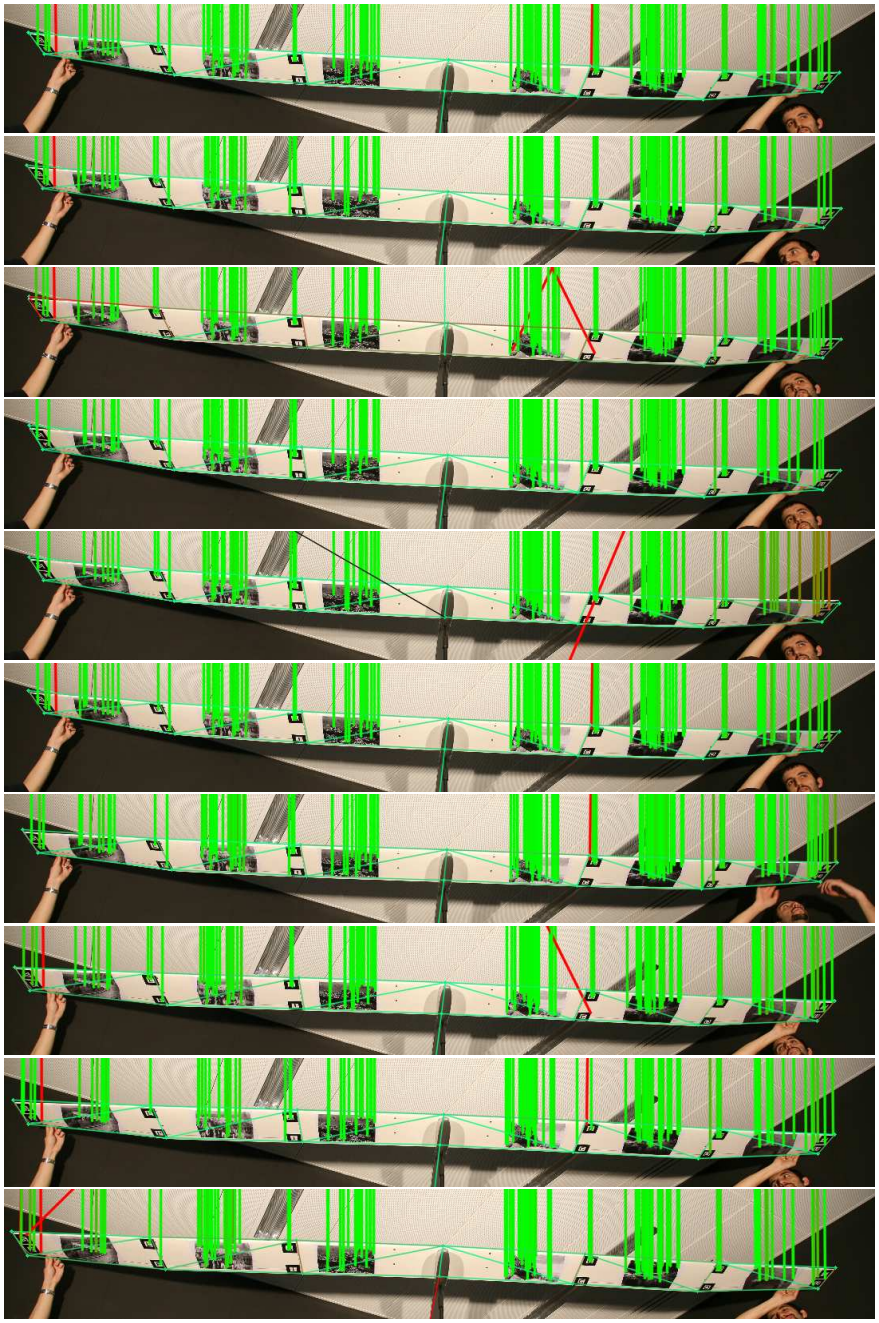


**Figure 6.6:** Estimated mesh reprojection in few samples of a video sequence.

# Bibliography


[1] P. Azzari, P. Fua, and P. Lagger. Video-based measurements of wing deformations. Technical report, CV Lab Tech Report, Ecole Polytechnique Federal Lausanne, Switzerland, 2008.

[2] R. I. Hartley B. Triggs, P. McLauchlan and A. W. Fitzgibbon. Bundle adjustment: A modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–372, 1999.

[3] D. A. Barrows. Videogrammetric model deformation measurement technique for wind tunnel applications. In *AIAA Aerospace Science Meeting and Exhibit*, 2007.

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *Proc. of European Conf. on Computer Vision*, pages 484–498, 1998.

[5] R. Hartley and A. Zisserman. *Multiple view Geometry in computer vision*. Cambridge University Press, Second Edition, 2003.

[6] Solar Impulse©. Around the world in a solar airplane.

[7] J.Pilet, V.Lepetit, and P.Fua. Fast non-rigid surface detection, registration and realistic augmentation. *Intl. Journal of Computer Vision*, 76(2):109–122, 2008.

[8] Y. W. Kwon and H. Bang. *The finite element method using MATLAB*. CRC Press, 2000.

[9] X. Llado, A. Del Bue, and L. Agapito. Non-rigid 3d factorization for projective reconstruction. In *Proc. of IEEE British Machine Vision Conference*, pages 484–498, 2005.

[10] I. Matthews and S. Baker. Active appearance models revisited. *Intl. Journal of Computer Vision*, 60(2):136–164, 2004.

[11] T. McInerney and D. Terzopoulos. A finite element model for 3d shape reconstruction and nonrigid motion tracking. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 518–523, 1993.

[12] D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and non-rigid motion tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.

[13] Photometrix©. Australis 7, http://www.photometrix.com.au.

[14] T. G. Ryall and C. S. Fraser. Determination of structural modes of vibration using digital photogrammetry. *AIAA Journal of Aircraft*, 39(1):114–119, 2002.

[15] M. Salzmann, J.Pilet, S.Ilic, and P.Fua. Surface deformation models for non-rigid 3d shape recovery. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(8):1481–1487, 2007.

[16] M. Salzmann, V. Lepetit, and P. Fua. Deformable surface tracking ambiguities. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[17] O. Urfalioglu, P. Mikulastic, and I. Stegmann. Scale invariant robust registration of 3d-point data and a triangle mesh by global optimization. In *Proc. of Intl. Conf. on Advanced Concepts on Intelligent Vision Systems*, pages 1059–1070, 2006.

[18] R. White and D. Forsyth. Combining cues: shape from shading and texture. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 1809–1816, 2006.

# Chapter 7

# Closing words

## 7.1 Summary

This thesis has investigated the problem of combining information contained in multiple, overlapping views of a scene for visual reconstruction purposes. Within this broad problem, three major topics have been addressed: dense geometric reconstruction , camera pose reconstruction, sparse geometric reconstruction of deformable surfaces.

**Dense geometric reconstruction.** Image mosaicing, the combination of several overlapping images into a collective view, has been the principal field of investigation. In this context, a robust and fast sequential image mosaicing algorithm has been conceived. By deploying novel spatial and tonal alignment approaches, the proposed method performs consistently in a wide range of real world scenarios, e.g. indoor and outdoor scenes.

An original dual geometric alignment stage permits to bound the drift error allowing the construction of quasi globally consistent mosaics, without resorting to computational demanding global adjustment procedures. The use of fast features, supplemented by a phase correlation based bootstrap, allows for handling large and complex camera motions while preserving real-time computation. A fast tonal alignment stage, based on histogram specification, has been conceived in order to deliver exact histogram matching and limited image distortion. Replacing standard mapping functions with one-to-many mapping relationships has been key to avoid histogram distortion artifacts without incurring in computationally intensive implementations.

Moreover, the mosaicing algorithm does not rely on any a priori information regarding scene or camera, thus resulting in a practical and flexible image-based solution. Accuracy, fast processing and flexibility have enabled integration into a video surveil-

lance system for on-line motion detection using a PTZ camera. Extensive experiments with several challenging photographs and surveillance sequences have shown the effectiveness of the proposed approach.

As far as principled performance assessment of mosaicing algorithms is concerned, to the best of our knowledge, no established evaluation framework exists in literature, albeit a widely accepted quantitative evaluation procedure is highly desirable for a discipline moving from its pioneering works to maturity. This issue has been addressed by devising a comprehensive evaluation methodology including data sets, ground-truth information and performance metrics. The effectiveness of the proposed methodology has been demonstrated by evaluating and ranking three algorithms that produce visually indistinguishable results.

**Camera pose reconstruction.** An original use of image mosaics in conjunction with standard pose reconstruction algorithms has been proposed. The idea is to model the reference object, i.e. the object with respect to which the pose is estimated, with a mosaic built offline from several detailed images. Standard pose reconstruction from planar object algorithms can then compute the pose between a given frame and the mosaic. Experiments, using two different pose estimation algorithms, have demonstrated considerable improvements in estimation accuracy. The mosaic-based pose reconstruction approach has been successfully integrated into a real-time Augmented Reality system under development in our Laboratory.

Moreover, a markerless vision-based approach based on natural features tracking has been conceived as a novel interface for gaming applications. The proposed approach allows the user to interact with a videogame by simply moving a webcam pointing towards any planar textured object present in the scene. The only requirements being a consumer grade camera, the proposed interface is practical, inexpensive and, according to the feedback received by several users, intuitive and enjoyable.

**Sparse reconstruction of deformable shapes.** A robust vision-based approach for accurate shape recovery of deformable surfaces from a single camera has been devised. Building on previous work in literature, the proposed method addresses the problem of obtaining highly accurate measurements of large and complex deformable objects, such as aircraft wings. State-of-the-art keypoints matching techniques have been deployed for non invasive, accurate and reliable sensing. A sophisticated modelization of the problem allows for dealing with reconstruction ambiguities, stemming from single view analysis, by introducing smoothness and continuity constraints in a concise way. A iterative linear LS estimation algorithm, based on projection kernels, delivers accurate results and fast computation.

A quantitative validation, using the SolarImpulse 4-meter wingspan scaled model, has reported reconstruction errors in the order of 2 mm compared to ground truth data, thus making it possible to foresee deployment of the method for accurate, cheap and non contact measurement of aircraft wings deformations.

## 7.2 Future directions

This final section discusses some possible avenues for future research and applications stemming from results and insights achieved in the course of the doctorate and discussed throughout this thesis.

**Evaluation methodology for image mosaicing algorithm.** For long time, image mosaics have been assessed subjectively via visual inspection, for qualitative applications such as digital photography, photomontage and post production effects, have been considered as the most important targets of such technology. The fast development in theoretical understanding, algorithms and processing power has rapidly raised the bar of mosaics quality to a level human eyes cannot discriminate or yield decisive insights. Moreover, nowadays mosaicing algorithms are employed not only to generate visually pleasant pictures but also serve as key building blocks of many computer vision applications, such as e.g. motion detection and tracking, mosaic-based localization, resolution enhancement, augmented reality. Finally, history teaches that the introduction of widespread accepted quantitative benchmarks invariably brought decisive benefits to the research within discipline, by facilitating communication, collaboration and dissemination among researchers dealing with similar challenges.

For these reasons, we hold a firm conviction that a widely accepted quantitative evaluation procedure is of utter importance for image mosaicing to moves from its pioneering works to maturity. The purpose of the evaluation methodology described in chapter 4is to provide the image mosaicing community with a comprehensive tool that, we hope, will allow for principled discussion about algorithms and performances among researchers and professionals. Data sets, rankings and further information on the evaluation methodology can be freely accessed at the web site `http://www.vision.deis.unibo.it/MosPerf`
All the researchers operating in the image mosaicing fields are heartily invited to use the methodology for evaluating their own algorithms, as well as to suggest insights, corrections, additional datasets or everything that could help improving our current proposal. The invitation is extended to companies developing commercial image mosaicing softwares, for they may gather useful insights by evaluating their commercial products, such as [6, 8, 10, 5, 3, 14, 11], according to the proposed methodology. Remarkably, no disclosure of any kind of technical detail is needed since just the mosaics

obtained on the reference data sets are required for the evaluation to take place.

**Vision-based interface for portable device games.** The ubiquitous presence of computerized equipments in everyday environment calls for conception and design of natural and easy-to-use human-machine interfaces. Practical, straightforward and inexpensive are the keywords for the next generation of interaction paradigms. Videogames are a challenging test ground since fast response and high accuracy are also required. Vision-based interfaces, as the one described in chapter 5.2, hold the potential to fulfill this expectation.

In particular, the segment of intelligent hand held devices, such smart-phones [2], PDA or consoles (Nintendo DS [7], Play Station Portable [12]), may see in the near future an ever-increasing penetration of vision based interface. Indeed, the proposed approach is particularly suited to enable gaming applications on hand held devices, for the user may simply point the integrated camera toward a textured plane and play by moving the device in his hand. Moreover, recent demonstration of camera pose reconstruction using natural keypoints on mobile phones allows for envisioning the deployment of camera-based games, such as Black Hole, on everybody's portable devices. Whatever the actual videogame, the proposed human-interface method may be employed as a general purpose middleware to deliver pose information, concerning the hand held device, to the game logic.

**Video-based metric measurement of dynamic scene.** Vision-based reconstruction approaches are known to recovery the geometric structure from the analysis of multiple views of the same subject. Several applications have already hit the market, e.g. ImageModeler [4], PhotoModeler [13], Boujou [1], Australis [9]. However, existing products are mainly intended for static scenes or dedicated to specific functions, i.e. image stabilization, super resolution. Moreover, the availability of a number of images may not be easily ensured in any given scenario.

The video-based measurement algorithm for deformable surfaces described in chapter 6 holds the potential to pave the way a new generation of accurate non invasive tools for geometric reconstruction of complex, static or dynamic, objects from single pictures, provided that a rest position model is available. Although the rest model has still to be constructed with traditional methods, once it is available shape reconstruction can be attained on-line from a single image and deformable objects or dynamic scenes can be handled seamlessly.

# Bibliography


[1] 2D3©. Imagemodeler, http://www.2d3.com/.

[2] Apple©. iphone , http://www.apple.com/iphone.

[3] ArcSoft©. Panorama maker 4, http://www.arcsoft.com/products/panoramamaker.

[4] Autodesk©. Imagemodeler, http://usa.autodesk.com.

[5] FirmTools©. Panorama composer 3, http://panorama.firmtools.com.

[6] Kolor©. Autopano pro 1.3, http://www.autopano.net.

[7] Nintendo©. Ds, http://www.nintendo.com/ds.

[8] Panavue©. Image assembler 3, http://www.panavue.com.

[9] Photometrix©. Australis 7, http://www.photometrix.com.au.

[10] Photovista©. Panorama 3, http://www.iseemedia.com.

[11] New House Software©. Ptgui 6, http://www.ptgui.com.

[12] Sony©. Play station portable, http://www.us.playstation.com/psp.

[13] EOS Systems©. Photomodeler, http://www.photomodeler.com.

[14] TekMate©. Photofit 1.4, http://www.photofit4panorama.com/.

# Bibliography


[1] 2D3©. Imagemodeler, http://www.2d3.com/.

[2] Apple©. iphone , http://www.apple.com/iphone.

[3] ArcSoft©. Panorama maker 4, http://www.arcsoft.com/products/panoramamaker.

[4] Autodesk©. Imagemodeler, http://usa.autodesk.com.

[5] P. Azzari. Robust image registration using linear and quadratic programming. Technical report, CV Lab Tech Report, University of Bologna, Italy, 2008.

[6] P. Azzari and A. Bevilacqua. Joint spatial and tonal alignment for motion detection with ptz camera. In *Proc. of Intl. Conf on Image Analysis and Recognition*, volume 4142, pages 764–775, 2006.

[7] P. Azzari, P. Fua, and P. Lagger. Video-based measurements of wing deformations. Technical report, CV Lab Tech Report, Ecole Polytechnique Federal Lausanne, Switzerland, 2008.

[8] P. Azzari, L. Di Stefano, F. Tombari, and S. Mattoccia. Markerless augmented reality using image mosaics. In *Proc. of Intl. Conf. on Image and Signal Processing*, 2008.

[9] R. I. Hartley B. Triggs, P. McLauchlan and A. W. Fitzgibbon. Bundle adjustment: A modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–372, 1999.

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

[11] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 1–8, October 2007.

[12] D. A. Barrows. Videogrammetric model deformation measurement technique for wind tunnel applications. In *AIAA Aerospace Science Meeting and Exhibit*, 2007.

[13] A. Bartoli, N. Dalal, B. Bose, and R. Horaud. From video sequences to motion panoramas. In *Proc. of Workshop on Motion and Video Computing*, pages 201–207, December 2002.

[14] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[15] A. Bevilacqua and P. Azzari. High-quality real time motion detection using ptz cameras. In *Proc. of IEEE Intl. Conf. on Advanced Video and Signal based Surveillance*, page 23, 2006.

[16] A. Bevilacqua and P. Azzari. A high performance exact histogram specification algorithm. In *Proc. of IEEE Intl. Conf. on Image Analysis and Processing*, pages 623–628, 2007.

[17] A. Bevilacqua, L. Di Stefano, and P. Azzari. An effective real-time mosaicing algorithm apt to detect motion through background subtraction using a ptz camera. In *Proc. of IEEE Intl. Conf. on Advanced Video and Signal based Surveillance*, volume 1, pages 511–516, 2005.

[18] A. Bevilacqua, L. Di Stefano, and A. Lanza. An efficient motion detection algorithm based on a statistical non parametric noise model. In *Proc. of IEEE Intl. Conf. on Image Processing*, pages 2347–2350, October 2004.

[19] K. S. Bhat, M. Saptharishi, and P. K. Khosla. Motion detection and segmentation using image mosaics. In *Proc. of Intl. Conf. on Multimedia and Expo*, volume 3, pages 1577–1580, 2000.

[20] P. Brodatz. Textures: a photographic album for artists and designers. In *Dover Publications*, 1999.

[21] L. Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[22] M. Brown and D. G. Lowe. Recognising panoramas. In *Proc. of the IEEE Intl. Conf. on Computer Vision*, pages 1218–1225, 2003.

[23] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *Intl. Journal of Computer Vision*, 74(1):59–73, August 2007.

[24] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. on Graphics*, 2:217–236, October 1983.

[25] F. M. Candocia. Jointly registering images in domain and range by piecewise linear comparametric analysis. *IEEE Trans. on Image Processing*, 12(4):409–419, 2003.

[26] D. Capel and A. Zisserman. Computer vision applied to super resolution. *IEEE Signal Processing Magazine*, 20(3):75–86, May 2003.

[27] D. P. Capel. *Image mosaicing and super-resolution*. University of Oxford, 2001.

[28] K. Cho, W. Kang, J. Soh, J. Lee, and H. S. Yang. Ghost hunter: a handheld augmented reality game system with dynamic environment. In *Proc. of Intl. Conf. on Entertainment Computing*, pages 10–15, 2007.

[29] B. Close, J. Donoghue, J. Squires, P. De Bondi, M. Morris, W. Piekarski, and B. Thomas. Arquake: an outdoor/indoor augmented reality first person application. In *Proc. of IEEE Intl. Symp. on Wearable Computers*, pages 139–146, 2000.

[30] D. Coltuc, P. Bolon, and J. M. Chassery. Exact histogram specification. *Trans. on Image Processing*, 15(5):1143–1152, May 2006.

[31] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *Proc. of European Conf. on Computer Vision*, pages 484–498, 1998.

[32] Intel©Corp. Opencv 1.0, open source computer vision library, 2000-2007.

[33] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, October 2003.

[34] R. Cucchiara, C. Grana, A. Prati, and S. Sirotti. Improving shadow suppression in moving object detection with hsv color information. In *Proc. of Intelligent Transportation Systems Conference*, pages 360–365, 2001.

[35] C. Dehais, M. Douze, G. Morin, and V. Charvillat. Augmented reality through real-time tracking of video sequences using a panoramic view. In *Proc. of Intl. Conf. on Pattern Recognition*, pages 995–998, 2004.

[36] A. Eden, M. Uyttendaele, and R. Szeliski. Seamless image stitching of scenes with large motions and exposure differences. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 2498–2505, 2006.

[37] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density for visual surveillance. In *Proc. of the IEEE*, volume 90, pages 1151–1163, July 2002.

[38] M. Eramian and D. Mould. Histogram equalization using neighborhood metrics. In *Proc. of Canadian Conf. on Computer and Robot Vision*, pages 397–404, 2005.

[39] M. Fiala. Artag, a fiducial marker system using digital techniques. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 590–596, 2005.

[40] FirmTools©. Panorama composer 3, http://panorama.firmtools.com.

[41] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381395, 1981.

[42] W. T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma. Computer vision for computer games. In *Proc. of Intl. Conf. on Automatic Face and Gesture Recognition*, pages 100–105, 1996.

[43] R. C. Gonzales and R. E. Woods. Digital image processing. *Upper Saddle River, NJ, Prentice Hall*, 2002.

[44] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*. Addison-Wesley, 2002. GON r 02:1 1.Ex.

[45] A. Govil, S. You, and U. Neumann. A video-based augmented reality golf simulator. In *Proc. of ACM Multimedia*, pages 489–490, 2000.

[46] M. D. Grossberg and S. K. Nayar. Determining the camera response from images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(11):1455–1467, November 2003.

[47] Khronos Group. Opengl 2.1, open computer graphics library, 1992-2008. `http://www.opengl.org/`.

[48] M. Grundland and N. A. Dogson. Color histogram specification by histogram warping. In *Proc. of SPIE Color Imaging X: Processing, Hardcopy, and Applications*, pages 610–621, January 2005.

[49] E.L. Hall. Almost uniform distributions for computer image enhancement. *IEEE Transactions on Computers*, 23(2):207–208, February 1974.

[50] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of Alvey Vision Conf,*, pages 147–151, 1988.

[51] R. Hartley and A. Zisserman. *Multiple view Geometry in computer vision*. Cambridge University Press, Second Edition, 2003.

[52] M. Harville and D. Li. Fast, integrated person tracking and activity recognition with plan-view templates from a single stereo camera. In *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition*, pages 398–405, 2004.

[53] D. Hasler and S. Susstrunk. Colour handling in panoramic photography. In *Videometrics and Optical Methods for 3D Shape Measurements*, pages 62–72, January 2002.

[54] E. Hayman and J. Eklundh. Statistical background subtraction for a mobile observer. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 67–74, 2003.

[55] R. Hess and A. Fern. Improved video registration using non-distinctive local image features. In *Proc. of IEEE Intl. Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[56] Solar Impulse©. Around the world in a solar airplane.

[57] M. Irani, P. Anandan, J.R. Bergen, R. Kumar, and S. Hsu. Efficient representations of video sequences and their applications. *Signal Processing Image Communication*, 8(4):327–351, May 1996.

[58] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Intl. Journal of Computer Vision*, 29(1):5–28, 1998.

[59] J.Pilet, V.Lepetit, and P.Fua. Fast non-rigid surface detection, registration and realistic augmentation. *Intl. Journal of Computer Vision*, 76(2):109–122, 2008.

[60] B. Brumitt K. Toyama, J. Krumm and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Proc. of Intl. Conf. on Computer Vision*, pages 255–261, 1999.

[61] S. Kang, J. Paik, A. Koschan, B. Abidi, and M. A. Abidi. Real-time video tracking using ptz cameras. In *Proc. of Intl. Conf. on Quality Control by Artificial Vision*, pages 103–111, May 2003.

[62] Alonzo Kelly. Mobile robot localization from large-scale appearance mosaics. *Intl. Journal of Robotic Research*, 19(11):1104–1125, 2000.

[63] S. J. Kim and M. Pollefeys. Radiometric self-alignment of image sequences. In *Proc. of IEEE Intl. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 645–651, 2004.

[64] Kolor©. Autopano pro 1.3, http://www.autopano.net.

[65] Y. W. Kwon and H. Bang. *The finite element method using MATLAB*. CRC Press, 2000.

[66] D. Lam. Tokamak, open physics engine library. `http://www.tokamakphysics.com/`.

[67] P. Liu, X. Sun, N. D. Georganas, and E. Dubois. Augmented reality: a novel approach for navigating in panorama-based virtual. In *Proc. Intl. Workshop on Haptic, Audio and Visual Environments and their Applications*, pages 13–18, 2003.

[68] X. Llado, A. Del Bue, and L. Agapito. Non-rigid 3d factorization for projective reconstruction. In *Proc. of IEEE British Machine Vision Conference*, pages 484–498, 2005.

[69] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of Intl. Conf. on Computer Vision*, pages 147–151, 1988.

[70] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60(2):91–110, 2004.

[71] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60(2):91–110, November 2004.

[72] P. Lu, Y. Chen, X. Zeng, and Y. Wang. A vision-based game control method. In *Proc. of Intl. Conf. on Computer Vision, Workshop on Human Machine Interaction*, pages 70–78, 2005.

[73] P. Lu, X. Y. Zeng, X. Huang, and Y. Wang. Navigation in 3d game by markov model based head pose estimating. In *Proc. of Intl. Conf. on Image and Graphics*, pages 493–496, 2004.

[74] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of Intl. Joint Conf. on Artificial Intelligence*, pages 674–679, April 1981.

[75] I. Matthews and S. Baker. Active appearance models revisited. *Intl. Journal of Computer Vision*, 60(2):136–164, 2004.

[76] T. McInerney and D. Terzopoulos. A finite element model for 3d shape reconstruction and nonrigid motion tracking. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 518–523, 1993.

[77] D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.

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

[79] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative o(n) solution to the pnp problem. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 1–8, October 2007.

[80] Nintendo©. Ds, http://www.nintendo.com/ds.

[81] Nintendo©. Wii. `http://wii.nintendo.com/`.

[82] Nasa© Earth Observatory. Picture of the day gallery.

[83] O. Oda, L. J. Lister, S. White, and S. Feiner. Developing an augmented reality racing game. In *Proc. of Intl. Conf. on Intelligent Technologies for Interactive Environment*, 2008.

[84] Panavue©. Image assembler 3, http://www.panavue.com.

[85] Photometrix©. Australis 7, http://www.photometrix.com.au.

[86] Photovista©. Panorama 3, http://www.iseemedia.com.

[87] F. Pitié, A. Kokaram, and R. Dahyot. Towards automated colour grading. In *Proc. of IEEE European Conference on Visual Media Production*, London, November 2005.

[88] F. Pitié, A. Kokaram, and R. Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, February 2007.

[89] PoV-Ray. Persistence of vision raytracer.

[90] A. Rosenfeld and A. Kak. Digital picture processing. *Upper Saddle River, NJ, Prentice Hall*, 1982.

[91] T. G. Ryall and C. S. Fraser. Determination of structural modes of vibration using digital photogrammetry. *AIAA Journal of Aircraft*, 39(1):114–119, 2002.

[92] S. Salti and L. Di Stefano. Svr-based jitter reduction for markerless augmented reality. In *Proc. of Intl. Conf. on Image Analysis and Processing (submitted paper)*, 2008.

[93] M. Salzmann, J.Pilet, S.Ilic, and P.Fua. Surface deformation models for non-rigid 3d shape recovery. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(8):1481–1487, 2007.

[94] M. Salzmann, V. Lepetit, and P. Fua. Deformable surface tracking ambiguities. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[95] M. Saptharishi, K. Bhat, C. Diehl, C. Oliver, M. Savvides, A. Soto, J. Dolan, and P. Khosla. Recent advances in distributed collaborative surveillance. In *Proc. of SPIE on Unattended Ground Sensor Technologies and Applications*, pages 199–208, April 2000.

[96] H. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *Proc. of European Conf. on Compuer Vision*, pages 103–119, 1998.

[97] H. S. Sawnhey and R. Kumar. True multi-image alignment and its applications to mosaicing and lens distortion correction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(3):235–243, March 1999.

[98] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Intl. Journal of Computer Vision*, 47(1-3):7–42, 2002.

[99] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(12):2024–2030, 2006.

[100] L.G. Shapiro and G.C. Stockman. *Computer Vision*. Prentice Hall Inc., New Jersey, 2001.

[101] H. Shum and R. Szeliski. Systems and experiment paper: construction of panoramic image mosaics with global and local alignment. *Intl. Journal of Computer Vision*, 36(2):101–130, 2000.

[102] G. Simon and M. Berger. Real time registration of known or recovered multi-planar structures: application to ar. In *Proc. of IEEE British Machine Vision Conference*, pages 567–576, 2002.

[103] G. Simon, A. W. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proc. of Intl. Symposium on Augmented Reality*, pages 120–128, May-June 2000.

[104] New House Software©. Ptgui 6, http://www.ptgui.com.

[105] Sony©. Play station portable, http://www.us.playstation.com/psp.

[106] Y. Sugaya and K. Kanatani. Extracting moving objects from a moving camera video sequence. In *Proc. of Symposium on Sensing via Image Information*, pages 279–284, June 2004.

[107] EOS Systems©. Photomodeler, http://www.photomodeler.com.

[108] R. Szeliski. Video mosaics for virtual environments. *Computer Graphics and Applications*, 16(2):22–30, 1996.

[109] TekMate©. Photofit 1.4, http://www.photofit4panorama.com/.

[110] C. Tomasi and J. Shi. Good features to track. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[111] Y. Uematsu and H. Saito. Vision-based registration for augmented reality with integration of arbitrary multiple planes. In *Proc. of Intl. Conf. on Image Analysis and Processing*, pages 155–162, 2005.

[112] O. Urfalioglu, P. Mikulastic, and I. Stegmann. Scale invariant robust registration of 3d-point data and a triangle mesh by global optimization. In *Proc. of Intl. Conf. on Advanced Concepts on Intelligent Vision Systems*, pages 1059–1070, 2006.

[113] P. Viola and M. J. Jones. Robust real-time face detection. *Intl. Journal of Computer Vision*, 57(2):137–154, 2004.

[114] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg. Towards massively multi-user augmented reality on handheld devices. In *Proc. of Intl. Conf. on Pervasive Computing*, pages 208–219, 2005.

[115] D. Wagner, G. Reitmayr, Alessandro Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In *Proc. of Intl. Symp. on Mixed and Augmented Reality*, pages 125–134, 2008.

[116] R. White and D. Forsyth. Combining cues: shape from shading and texture. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 1809–1816, 2006.

[117] F. Winkelman and I. Patras. Online globally consistent mosaicing using an efficient representation. In *Proc. IEEE Intl. Conf. on Systems, Man and Cybernetics*, pages 3116–3121, October 2004.

[118] Y. Xiong and K. Turkowski. Registration, calibration and blending in creating high quality panoramas. In *Proc. of IEEE Workshop on Applications of Computer Vision*, pages 69–74, 1998.

[119] Y. J. Zhang. Improving the accuracy of direct histogram specification. *Journal of Electronic Imaging*, 28(3):213–214, 1992.

[120] Z. Zhu, G. Xu, E. M. Riseman, and A. R. Hanson. Fast generation of dynamic and multi-resolution 360 panorama from video sequences. In *Proc. IEEE of Intl. Conf. on Multimedia Computing and Systems*, pages 400–406, July 1999.

[121] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.

[122] S. Zokai and G. Wolberg. Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. *IEEE Trans. on Image Processing*, 14(10):1422–1434, October 2005.