Università degli Studi di Pisa
Settore Scientifico Disciplinare: INF/01

Ph.D. Thesis

Advancements in multi-view processing for reconstruction, registration and visualization.

Luca Benedetti

Supervisor
Dr. Roberto Scopigno

Supervisor
Dr. Massimiliano Corsini

Referee
Prof. Luc Van Gool

Referee
Prof. Michael Wimmer

December, 2013

Abstract

The ever-increasing diffusion of digital cameras and the advancements in computer vision, image processing and storage capabilities have lead, in the latest years, to the wide diffusion of digital image collections. A set of digital images is usually referred as a *multi-view* images set when the pictures cover different views of the same physical object or location.

In multi-view datasets, correlations between images are exploited in many different ways to increase our capability to gather enhanced understanding and information on a scene. For example, a collection can be enhanced leveraging on the camera position and orientation, or with information about the 3D structure of the scene. The range of applications of multi-view data is really wide, encompassing diverse fields such as image-based reconstruction, image-based localization, navigation of virtual environments, collective photographic retouching, computational photography, object recognition, etc. For all these reasons, the development of new algorithms to effectively create, process, and visualize this type of data is an active research trend.

The thesis will present four different advancements related to different aspects of the multi-view data processing:

- *Image-based 3D reconstruction*: we present a pre-processing algorithm, that is a special color-to-gray conversion. This was developed with the aim to improve the accuracy of image-based reconstruction algorithms. In particular, we show how different dense stereo matching results can be enhanced by application of a *domain separation* approach that pre-computes a single optimized numerical value for each image location.

- *Image-based appearance reconstruction*: we present a multi-view processing algorithm, this can enhance the quality of the color transfer from multi-view images to a geo-referenced 3D model of a location of interest. The proposed approach computes virtual shadows and allows to automatically segment shadowed regions from the input images preventing to use those pixels in subsequent texture synthesis.

- *2D to 3D registration*: we present an unsupervised localization and registration system. This system can recognize a site that has been framed in a multi-view data and calibrate it on a pre-existing 3D representation. The system has a

very high accuracy and it can validate the result in a completely unsupervised manner. The system accuracy is enough to seamlessly view input images correctly super-imposed on the 3D location of interest.

- *Visualization*: we present *PhotoCloud*, a real-time client-server system for interactive exploration of high resolution 3D models and up to several thousand photographs aligned over this 3D data. PhotoCloud supports any 3D models that can be rendered in a depth-coherent way and arbitrary multi-view image collections. Moreover, it tolerates 2D-to-2D and 2D-to-3D misalignments, and it provides scalable visualization of generic integrated 2D and 3D datasets by exploiting data duality. A set of effective 3D navigation controls, tightly integrated with innovative thumbnail bars, enhances the user navigation.

These advancements have been developed in *tourism* and *cultural heritage* application contexts, but they are not limited to these.

Acknowledgments

I want to thank all the wonderful people who made this work possible.

Thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we introduce the multi-view data and its vast application fields. Multi-view data is increasingly gaining importance in diverse application fields related to the digitalization of real-world objects and locations.

1.1 Motivation

The diffusion of digital cameras in the recent years (e.g. mobile phones, tablets, etc.) has created new scenarios related to the digitalization of the world around us. Digital collection of images are becoming one of the most diffused user data. When images are related to different views of a physical location or object, we can say that this collection is a *multi-view* image set, or a *multi-view* datum. This data is usually more useful if enriched with extra information such as: camera parameters (implicit or/and explicit), the three dimensional shape of the subject location, etc.

In latest years, many different research trends have emerged in order to find algorithms to effectively create, process, visualize, and in general exploit this type of data. This thesis presents some advances in these areas, with focus on the author contributions.

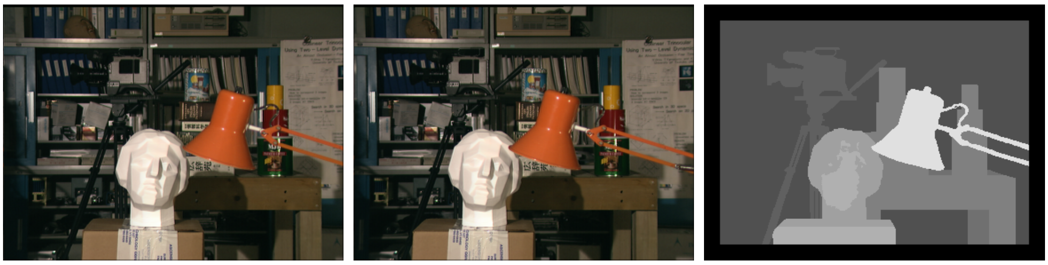


Figure 1.1: Left and center: a pair of stereo images depicting the same scene (Tsukuba dataset [282]). Right: the disparity map between the input images is an example of data that can be inferred from them.

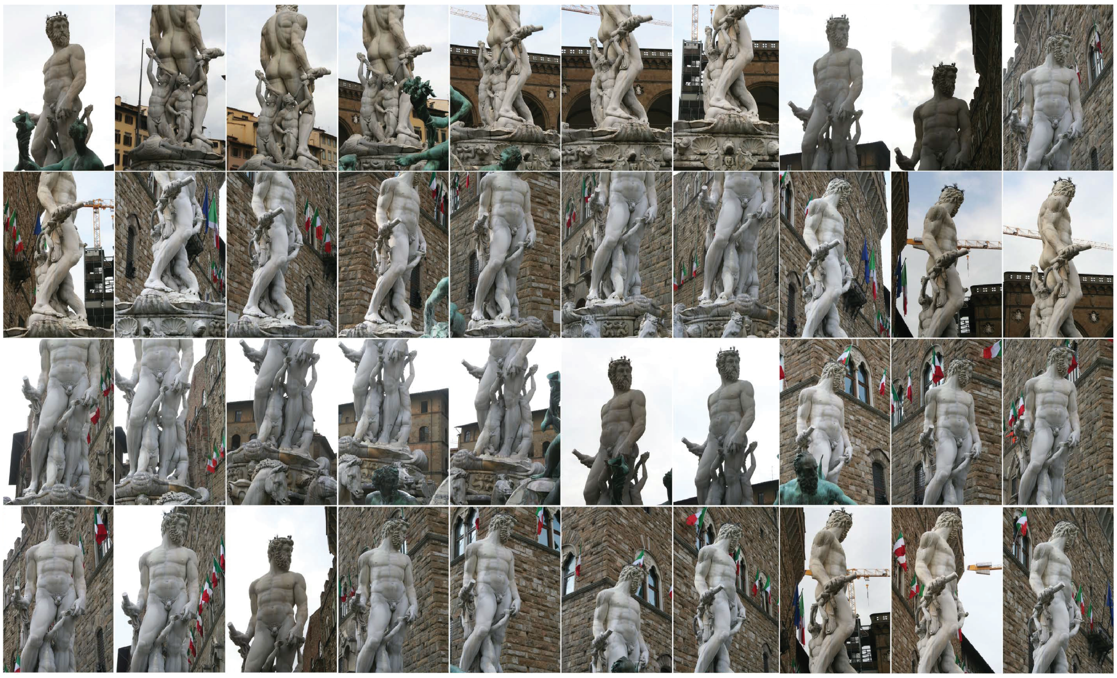


Figure 1.2: A multi-view image set; this was captured in a dedicated outdoor campaign to acquire the appearance of the *Fountain of Neptune* statue in *Piazza della Signoria* in Florence, Italy.

In general, we can define multi-view data as *any* kind of “same location”-related imaging set. Under this definition, different cases are encompassed:

- *stereo pairs*, the simplest case, a pair of images taken from a stereo-rig, as shown in Fig. 1.1;
- *dedicated photographic shots*, the photographs are intentionally shot for a “multi-view” purpose. The data set typically covers as much as possible the reachable scene/object surface, and similar camera settings are used between shots. An example can be seen in Fig. 1.2. The images in a dataset can range from controlled in-lab setups to outdoor acquisition campaigns without any assumption in the *ordering* of the images.
- *web-retrieved collections* refers to uncontrolled datasets retrieved through a textual web query. This is the most complex case for still images, because images can be extremely varied, and “noise” can be present in the form of people, objects or image edits. An example is shown Fig. 1.3. These datasets can have false positives; that are images not related to the set. This issue requires the need of coherency checks and scalability; due to and the high amount of images in the typical collection.
- *frames of a video sequence*, if the depicted scene is static and the camera

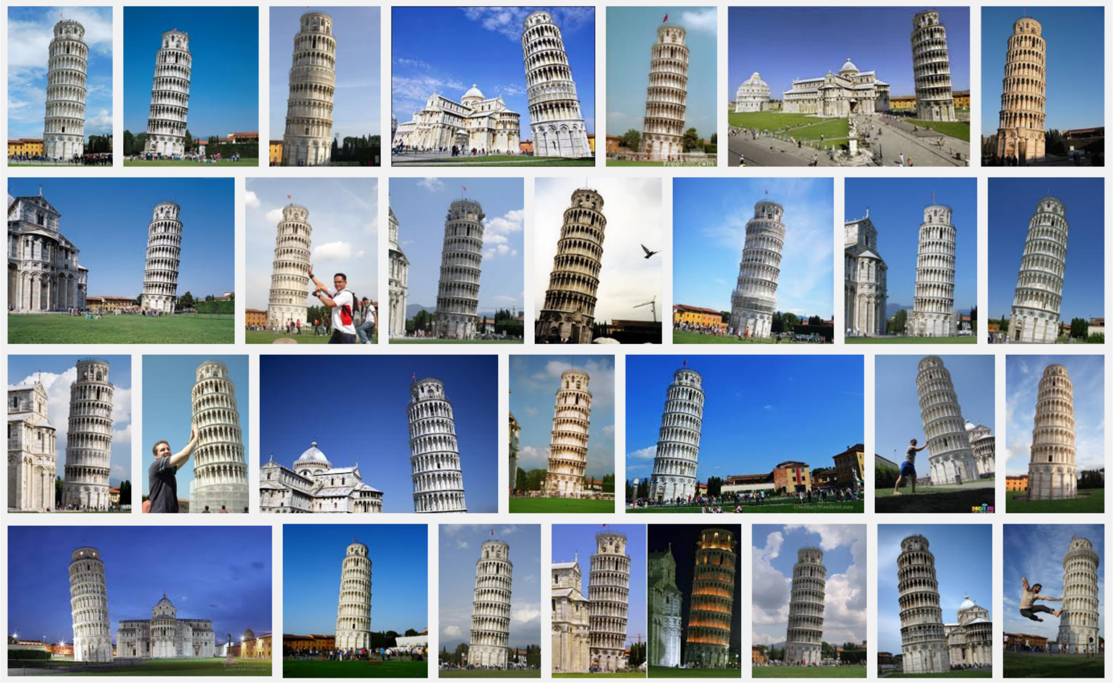


Figure 1.3: Part of the results obtained by searching "leaning tower of Pisa" on Google Images.

position moves in or around it, video sequences can be used as multi-view data sets for which a *temporal/spatial continuity assumption* can be made; see Fig. 1.4;

- *multiple videos streams* can be used to capture a dynamic scene in a multi-view fashion; see Fig. 1.5.

This variety leads to radical differences between the assumptions that can be made on the data and on the amount of information that can be extracted from it. In this thesis, we focus on the first three categories, i.e. the ones concerning still images.

1.2 Multi-view applications

The fact that a scenario, or an object, is depicted from more than a single point of view poses both new challenges and opportunities for Computer Vision, Computer Graphics and Image Processing. Typically, the images need to be put in correspondence between each other in some way. These relationships open far stronger assumptions that can be made on the scene/object to achieve some tasks with respect to traditional single image processing. Many classical single-image approaches have been re-thought in this context in order to benefit from the exploitation of multi-view data.

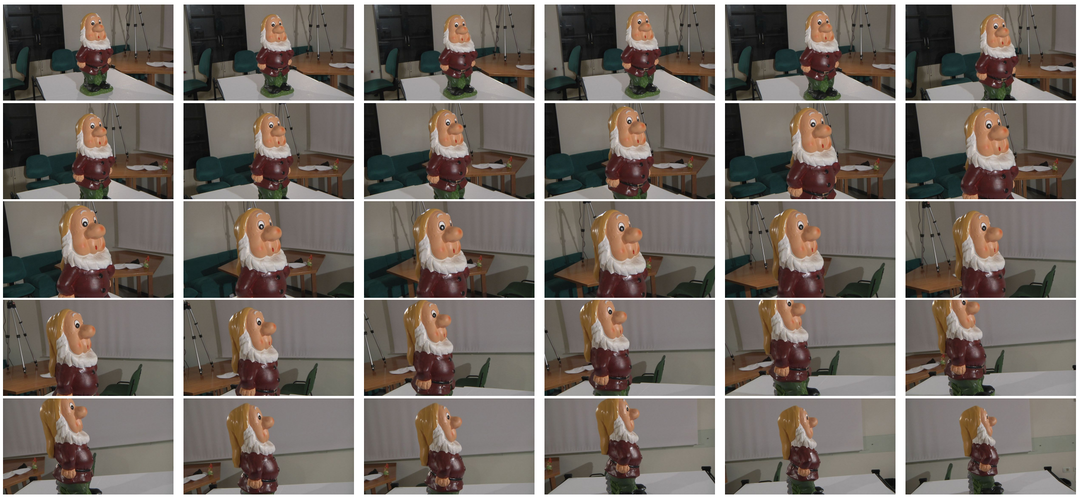


Figure 1.4: Temporal and spatial continuity between neighbors in a video sequence: left to right, then top to bottom: frames extracted from a video stream of an in-lab acquisition setup [248].

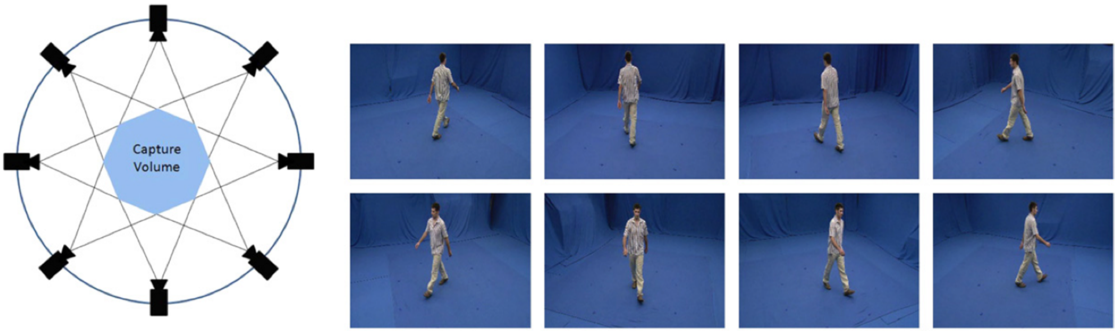


Figure 1.5: Left: multi-video controlled setup for human movement recognition. Right: an example of the captured frames (from Iosifidis et al. [166]).

Image-based scene reconstruction, localization, registration, rendering, and visualization are inherently multi-view based fields and are covered in Chapter 2, due to their relevance with the thesis contributions. Other than those, the fields of application in which multi-view image data show to be beneficial are:

- *Object recognition and classification*, where systems are increasingly exploiting muti-view datasets to improve accuracy and recall on face detection [140, 196, 368], object detection [372], recognition [101, 349], and class detection [330]. There are also compact multi-view descriptors for 3D object retrieval [78] that use view-based approaches for 3D visual features object retrieval. Bag-of-words approaches [63] can use the geometrical relationships of multi-view data to better validate results.
  Other cases in which multi-view approaches have been demonstrated to be

very useful are deformable shape matching [197], viewpoint classification and synthesis of object categories [316], robotic recognition [331, 350], traffic sign recognition and 3D localization [332], Hough transform based object detectors [267], action recognition and pose estimation [365], and image classification [324].

- In the *surveillance and tracking* fields, systems have been integrating multiple video sources [29, 242], in order to improve tracking robustness against occlusions and unseen parts of the scene, enable marker-less body tracking [175], monitor traffic [203], and identify gait [162, 141].

- The applicability of multi-view data has been recently proven in the context of *medical imaging.* For example, as a component that can be integrated in real-time 3D echocardiography [264] and mammographic mass retrieval systems [204].

- Multi-view data redundancy presents many opportunities for optimizing *general-purpose image processing* algorithms which are usually applied to single images. Early approaches dealt with the problem of extending image compression to multi-view sets to exploit the intrinsic redundancy of the data in a domain-specific way, by basing the coding on affine transforms [115], inter-frame correlation [10], or geometrical coding [124]. After compression, many other usage cases emerged such as: digital matting [165], denoising [369, 364], computation of quality measures [311], color correction [363], brightness correction [373], intrinsic image computation [187], and user-assisted image compositing [39].
  The latest trend has been to expand multi-view techniques for image processing to *dynamic* scenes, e.g. a scenario in which people moves around between shots. In this case, rigid correlation between images is not possible. However, partial correlations between parts of the images still allow, for example, to optimize color consistency of a photo collection by acting simultaneously on the images [143].

- Due to temporal requirements, *multi-view video analysis* is in general more complex with respect to multi-image analysis. Nevertheless, this field has recently gained popularity. This is because it allows for specialized solutions in human pose estimation [153], and human movement recognition [166], but also for consistent color correction across different video streams [291]. Furthermore, a natural extension to image-based rendering is video-based rendering and video-based animation, that uses one or more videos in order to create novel video-based experiences [284].
  Related to this area, 3D videos created from multiple video streams [297, 209] and video-based breakthroughs of environments [338] have widespread applications in driving direction systems and immersive outdoor mapping.

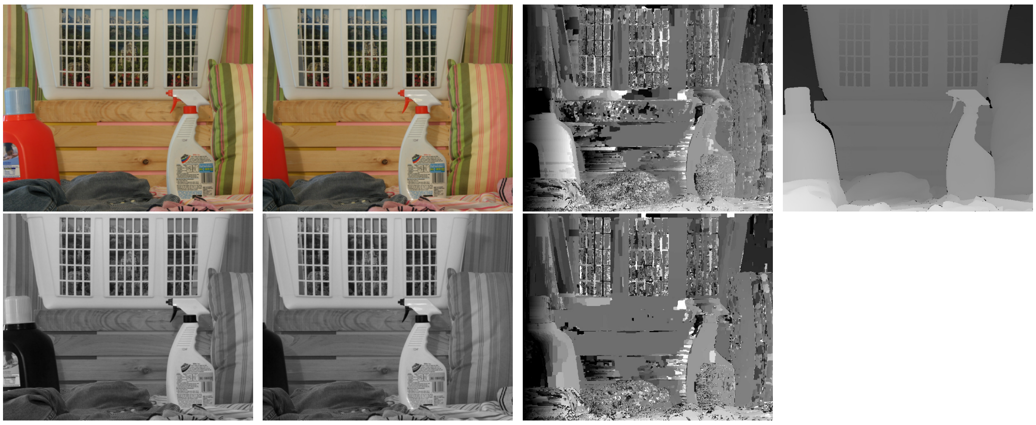


Figure 1.6: Top line: a color stereo pair, a dense stereo matching result and the disparity map ground truth. Bottom line: the same image pair, preprocessed with our proposed system and the resulting dense stereo matching result. Even if it is not easily visible in the disparity maps, an error measure of the color version (61.65%) with respect to the ground truth decreases significantly (to 48.13%) just by applying the proposed gray-scale preprocessing.

1.3 Contribution

In this thesis, we present four different advancements related to four different aspects of multi-view processing, that cope with issues lying both in Computer Vision, Computer Graphics and Image Processing. The main contributions presented are summarized in the following.

1.3.1 Image-based 3D reconstruction

In Sec. 2.1, we introduce the image-based 3D reconstruction. A common trait in this field is that, in order to produce a 3D surface from sets of image pixels, a single numerical value for each pixel is often needed, for example to calculate matching costs or to extract features. In most scenarios, images are shot in color, and existing Computer Vision algorithm implementations usually perform simple aggregation of those color values.

In this regard, we present an image pre-processing algorithm, that is a special color-to-gray conversion, that helps in improving the accuracy of general image-based reconstruction algorithms by means of a *domain separation* strategy. The proposed work does not address directly a specific Computer Vision algorithm such as a feature matching system or a depth map estimator. Instead, it explores the space of image pre-processing techniques in order to optimize the input for general classes of 3D reconstruction techniques.

The aim is to understand the conversion qualities that can improve the accuracy of results when the gray-scale conversion is applied as a pre-processing step in the context of vision algorithms, and in particular dense stereo matching. We test and show results in this specific case, showing how matching results can be enhanced in different dense stereo algorithms by pre-computing a single optimized numerical value for each image location, as anticipated in Fig. 1.6.

In order to achieve this result, we perform a study of advanced color to gray conversion approaches. These are usually designed for human preference, but we focus on the theoretical aspects that are relevant for computer vision-related performance. Between those, we can mention *feature discriminability*, to be able to perform matching even at the cost of human-perceptual worsening of the quality; *chrominance awareness*, to be able to discriminate between iso-luminant colors; and *color consistency* to map the same color to the same gray-scale value in every image even if the conversion depends on the image contents. Then, we select the most relevant approach, and we enhance it to fulfill all our requirements. The system is then thoroughly tested with a standard and well-known dense stereo matching framework, in order to compare its results with both color processing and other gray-scale preprocessing approaches.
The publication relative to this contribution is:

> Benedetti Luca, Corsini Massimiliano, Cignoni Paolo, Callieri Marco, and Scopigno Roberto.
> *Color to gray conversions in the context of stereo matching algorithms. An analysis and comparison of current methods and an ad-hoc theoretically-motivated technique for image matching.*
> In Machine Vision and Applications, pages 1–22, Springer Berlin / Heidelberg, see reference [22]

1.3.2 Image-based appearance reconstruction

In Sec. 2.1.4, we introduce the image-based *appearance* reconstruction. Once a 3D model geometry has been put in relation with a relative multi-view image set, the color from the images can be transferred to the model surface in order to approximate the object appearance. This transfer is not free of approximations and defects. One of the most widely spread problems is the presence of shading artifacts in the images such as shadows and highlights. These alter the synthesized appearance of the object if transferred as-is.

In controlled environments, the lighting can be optimized, but this is not possible in outdoor acquisition campaigns, and the best possible approximation is to shot photographs on a cloudy day in order to minimize artifacts. In some parts of the world, e.g. in the tropics, cloudy days are rare and there is almost a certainty of having strong shadows in images.

Once again, we thought of a pre-processing approach to mitigate this issue. Thus, we present a 3D-based image shadow removal algorithm, that enhances the quality of the color transfer from the calibrated images to the geo-referenced 3D model.

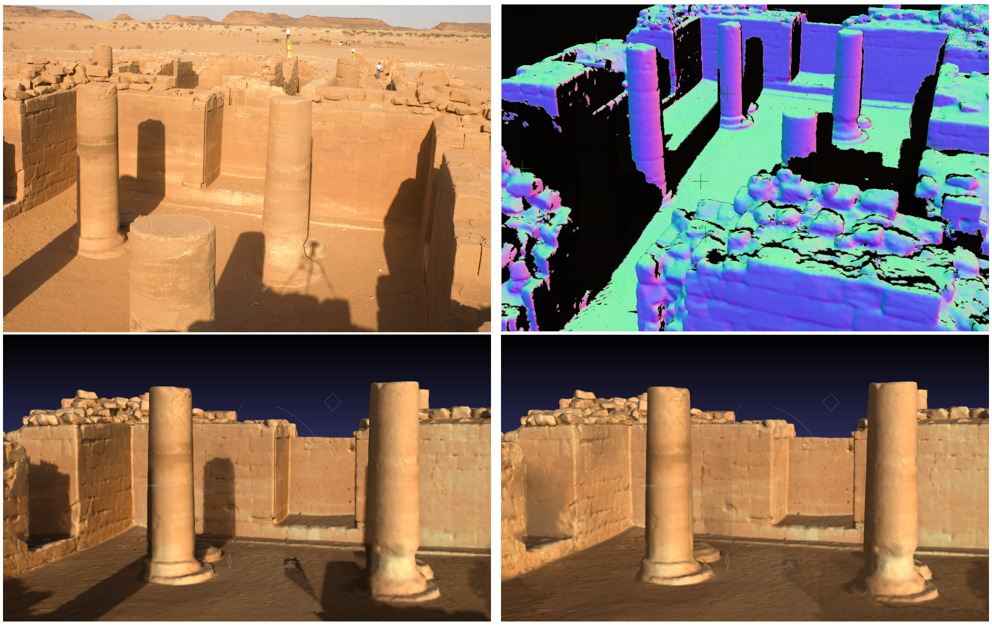


Figure 1.7: Top left: an image with strong sun light. Top right: the rendering of the corresponding 3D model with normal maps and shadows generated by the estimated sun light direction. Bottom left: a rendering of the 3D model with the unaltered images used for coloring. Bottom right: the same rendering, with the proposed shadows removal pre-processing.

The approach computes virtual shadows using a re-computation of the sun position, segments shadowed regions from the input images, then assign a “bad quality” to shadowed regions to prevent use of inciding pixels in subsequent texture synthesis when possible. Furthermore, it removes the shadows from the input images in order to gracefully provide color data where the only color source for part of the surface comes from shadowed regions. Fig. 1.7 shows an overview of the system.
The publication relative to this contribution is:

> Dellepiane Matteo, Benedetti Luca, and Scopigno Roberto.
> *Removing shadows for color projection using sun position estimation.*
> In 11th VAST International Symposium on Virtual Reality, Archaeology and Cultural Heritage, page 55–62, Eurographics, see reference [87]

1.3.3 Large scale 2D/3D registration

In Sec. 2.2, we introduce image registration and image localization. Established 2D/3D registrations approaches are focused in increasingly automatic very precise calibration of photographs with respect to known 3D surfaces. Traditional image-based localization approaches focus in expanding a multi-view dataset with new images. Typically, registration is performed at small scales with single images, on

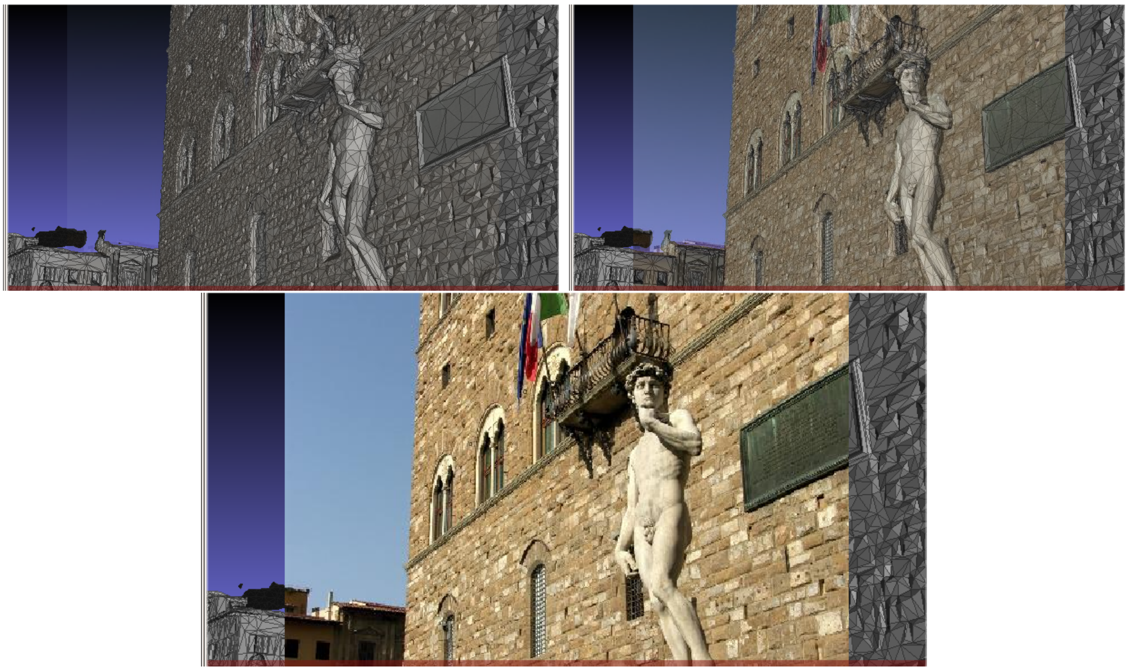


Figure 1.8: Alignment results at increasing opacity levels, showing the accuracy of the proposed unsupervised system.

the contrary, localization is characteristically done at large scales, but with very low precision. Moreover, both processes are usually supervised, i.e. they minimize an error function. However, human resources often need to check the final result. This is usually not a problem in registration, because the multi-view datasets used tend to be not too large. However, it can be a problem in localization scenarios in which huge datasets are exploited.

We contribute on all these issues at the same time. We propose a system which is able to combine an *unsupervised* image-based localization technique with an automatic 2D/3D registration approach. Given a database of multi-view data sets with corresponding 3D models, the system can recognize if a new input image belongs to one of the sites, and calibrates it on the corresponding 3D representation. In order to achieve this, we first use an advanced large-scale 2D/2D matching approach to select the right multi-view dataset from the available ones, and specifically the most relevant images. Then, we extract 2D/2D correspondences between these images, compute a calibration using the correspondences, the 3D model and the already existing support calibrations. We finalize the calibration by comparing its results with independent Structure and Motion calibration.

The system main strengths are the completely unsupervised nature, its very high accuracy. This is high enough to seamlessly view the input image correctly super-imposed on the 3D location of interest, as can be seen in Fig. 1.8.

In theory, an extension of the color to gray algorithm mentioned above, that

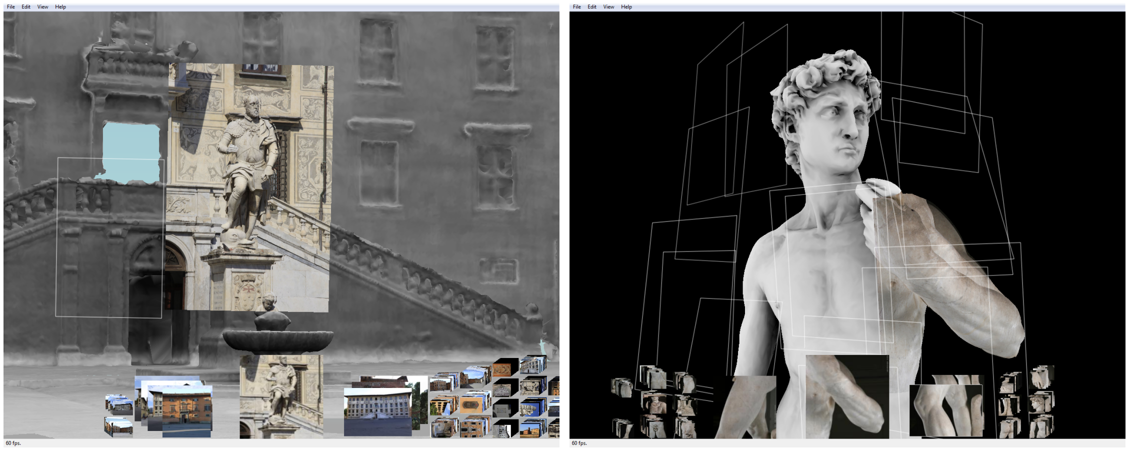


Figure 1.9: Examples of the proposed visualization system. Left: navigation in *Piazza dei Cavalieri* in Pisa, Italy, showing an image immersed in the 3D scene and the framelets of other nearby images in the multi-view data set. Right: visualization of the Michelangelo's David, using a projective texture mapping to color the surface relative to the nearest image with respect to the observer position.

is ad-hoc for multi-view reconstruction scenarios, could be used here for improving results by supporting the reconstruction phase for robustness. We did not test such extension mainly due time constraints.
The publication relative to this contribution is:

> Benedetti Luca, Corsini Massimiliano, Dellepiane Matteo, Cignoni Paolo, and Scopigno Roberto.
> *GAIL: Geometry-aware Automatic Image Localization.*
> In VISAPP 2013 - International Conference on Computer Vision Theory and Applications, Number in press - 2013, see reference [23]

1.3.4 Visualization and navigation

In Sec. 2.3, we introduce the visualization of multi-view data. An important issue in these visualization systems is related to the *amount* of data itself: the image collections can be huge and the 3D models can be at really high resolutions. Those size problems are mainly efficiency-related for the 3D part, but there are, in addition, strong interface-effectiveness issues for the 2D part. An interactive multi-view visualization system has to exploit intuitive and effective image arrangements into the available screen space, also shared with the 3D model rendering. Moreover, the visualization system needs to be memory-efficient with respect to the gigabytes of data usually reached by non-trivial multi-view datasets.

Another problem is that 3D models are not guaranteed to be complete. This is due to occluding objects or warped surfaces, even if they represent non-occluded surface geometry with a really high precision. Color enrichment of the visible surfaces can be either be managed with heavy sampling and point/vertex coloring, that is impractical, texture mapping, that is quite tedious to achieve or direct color projection from calibrated images. However, these approaches require perfect 2D/3D calibrations and coherency between nearby images in order to avoid various kind of artifacts. These expectation of perfection are not realistic in the general case, and a dynamic solution is needed to reliably associate image-derived colors to the geometrical surfaces.

To front these issues, we present *PhotoCloud* (see Fig. 1.9), a real-time client-server system for interactive exploration of high resolution 3D models and up to several thousand photographs aligned over this 3D data. The system improves over current state of the art by supporting any kind of 3D model that can be rendered in a depth-coherent way (point clouds, triangle soups, and indexed triangle meshes) and arbitrary multi-view image collections. It provides scalable visualization of generic integrated 2D and 3D datasets, and tolerates 2D-to-2D and 2D-to-3D misalignments.

A set of effective 3D navigation controls, tightly integrated with innovative thumbnail bars, enhance user navigation of the data. For example, spatial ordering can be imposed on the thumbnails in order to provide relevant images and clustering approaches are used to present the maximum amount of image information in the limited screen space available.

This system has been extensively proved in synergy with the large scale 2D/3D registration system mentioned above in the context of a successful research project for interactive tourism applications. Due to these two systems, users have been able to visualize their photographs contextualized in a 3D scene of their visits, without any manual data processing.
The publication relative to this contribution is:

> Brivio Paolo, Benedetti Luca, Tarini Marco, Ponchio Federico, Cignoni Paolo, and Scopigno Roberto.
> *PhotoCloud: Interactive Remote Exploration of Joint 2D and 3D Datasets.*
> In IEEE Computer Graphics and Applications, vol. 33, no. 2, pp. 86-96, c3, March-April 2013, see reference [45]

1.4 Thesis structure

The thesis is organized as follows.

Chapter 2 provides a general overview about the state of the art of the research areas involved in multi-view processing of our interests. The field treated are: image-based 3D reconstruction, image-based appearance reconstruction; image-based localization and image-based visualization and navigation.

Chapter 3 presents our contribution in multi-view processing for image-based 3D reconstruction. Here, the proposed ad hoc color aggregation algorithm to improve image matching is described. An in-deep background about color-to-gray conversions is also given.

Chapter 4 presents our contribution in multi-view processing for image-based appearance reconstruction. The shadow removal algorithm for outdoor scene proposed to improve color acquisition of large 3D objects/structures is detailed in this chapter.

Chapter 5 presents our contribution in multi-view processing for 2D/3D registration. Our very accurate image-based localization algorithm, recast as a large 2D/3D registration problem, is detailed here.

Chapter 6 presents our contribution in the visualization of multi-view data. An in-deep description and analysis of the PhotoCloud visualization system developed during the work of this thesis is presented.

Finally, Chapter 7 outlines the conclusions about the proposed solutions, and provides the list of publications produced by the main contributions of the thesis.

Chapter 2

Related work

In order to provide context for the following chapters, we introduce the related work on treated topics. Related work is provided for the main steps of image-based 3D reconstruction, for appearance reconstruction, for localization and registration and for visualization of multi-view data.

2.1 Related work on image-based reconstruction

The most prominent field of application of multi-view data is the reconstruction of the scene depicted in the pictures.

Three dimensional reconstruction is the process of recovering the properties of an environment and, optionally, characteristics and parameters of the sensing instrument from a series of measures. This generic definition is broad enough to accommodate very diverse methodologies such as: time-of-flight (ToF) laser scanning, photometric stereo or satellite triangulation. However, we focus here on algorithms and techniques to resolve the problem of reconstructing three-dimensional scenes using only photographic information. Traditionally, most of the literature partitions this complex task into three main steps:

Image Matching: in which algorithms compute accurate correspondences between parts of different photographs.

Structure and Motion: that uses extracted correspondences to estimate both intrinsic and extrinsic camera parameters, and recover 3D sparse points of the depicted scene/object as a side result.

Multi-view Dense Stereo: that takes images with calibrated cameras as input and produces dense 3D models.

In the last few years, the mix of such technologies has allowed the Computer Vision community to reconstruct increasingly bigger and more complex scenarios, generally using decreasing amounts of auxiliary information. As of today, a completely general reconstruction algorithm, that guarantees bounds on the errors, does not exists even though current results are impressive. For example, research projects succeeded in reconstructing (with some errors) even large portions of cities using millions of images gathered from the Internet. Similarly, single architectural elements [224] can be, in many cases, densely extracted and reconstructed with the latest technologies.

The progress in the 3D reconstruction research has been rapid. This has been fueled by the interest of industry and general public, the advances in computational power of desktop and mobile devices, the advent of wide diffusion of digital photography and the subsequent availability of large dataset of public images, the recent breakthroughs in key point matching, auto-calibration and multi-view dense reconstruction algorithms treated in many widely diffused Computer Vision books [15, 104, 98, 231, 334, 108, 319]. A few years ago, research in this field was only dealing with controlled dataset composed by few images; e.g. stereo-rig or an image set obtained using a turn-table. However, the efforts toward this goal trace a long way back, and literature in this field is now vast. The current state of the art is the result of at least 50 years of efforts in this direction, evolving from early attempts to identify lines in images, label them and infer a “block world” structure from the topological connections [270]. From there, a huge amount of work has been brought forward, and describing it all would be out of the scope of this thesis. The main related techniques for which we will not go into detail are listed below, with reference to some of the most significant papers and surveys:

- Line labeling and edge detection [163, 70, 346, 273, 171, 80]
- Optical flow [160, 208, 227, 228, 36, 161, 6, 24, 30, 49, 249, 227, 18, 13]
- Image pyramids [272, 51, 52, 104, 6, 353, 354, 201] and wavelets [1, 211, 300, 301, 302]
- “Shape-from-X” techniques [155, 250, 33, 158, 159, 358, 352, 250, 210, 234, 156]
- Variational optimization problems [327, 255, 328, 34, 27, 329]
- Markov Random Field models [125, 254, 253, 105, 322, 32] and Kalman filter models [91, 215, 317]
- Physics-based approaches [356, 148, 290]
- Graph cuts [42, 182, 176] and loopy belief propagation [366, 185]

In the next section, we describe in details the three main steps in image-based reconstruction just mentioned. Firstly, we focus on the image matching aspect. Secondly, we introduce the Structure and Motion and the Multi View Stereo aspects. Finally, we describe the reconstruction of the scene/object or *appearance*.

2.1.1 Image matching

Image Matching is an essential component of many computer vision applications that copes with the detection of salient image features; with the description of such features and the coherent matching of the most similar features between different photos of the same scene.

A *feature detector* extracts the interesting parts of the image, usually points or lines. A good detector should be repeatable and reliable. Repeatability means that the same feature can be detected in different images. Reliability means that the detected point should be distinctive enough so that the number of its matching candidates is small.

A *feature descriptor* associates to each extracted feature a descriptive information. This is usually presented in form of vector reused in the matching process. A descriptor should be invariant to rotation, scaling, and affine transformation so the same feature on different images will be characterized by almost the same value.

Features can be points, edges or areas, depending on the used techniques. The descriptions and the matching techniques are also highly variable. Point features can be used to find a sparse set of corresponding locations in different images, often as a precursor to computing camera pose, which is a prerequisite for computing a denser set of correspondences using stereo or multi-view stereo matching.

Such correspondences can also be used to align different images; e.g. stitching image mosaics or performing video stabilization. They are also used extensively to perform object instance and category recognition.

The most important advantage of key points is that they permit matching, even in the presence of clutter, occlusions, large scale differences and orientation changes. Feature-based correspondence techniques have been used since the early days of stereo matching [144, 90, 213, 221, 214, 12, 17, 222, 244, 137, 256, 262, 145, 245, 172, 41].

There are two main approaches for finding feature points and their correspondences. The first is to find features in one image that can be accurately tracked using a local search technique, such as correlation or least squares. The second is to independently detect features in all images under consideration and then match features based on their local appearance. The former approach is more suitable when images are taken from nearby viewpoints or in rapid succession; e.g. video sequences. On the other hand, the latter is more suitable when a large amount of motion or appearance change is expected; e.g. in establishing correspondences in wide baseline stereo [279].

The Harris corner detector [146] is a well-known point detector and it is invariant to rotation and partially to intensity change. However, it is not scale invariant. The detector is based on a local auto-correlation function that measures the local changes of the image.

Scale invariant detectors [205, 217] search for image features over scale and space. SIFT [205] searches for local maxima of Difference of Gaussian (DOG) in space and scale. Mikolajczyk and Schmid's method [217] exploits Harris corners to search for features in the spatial domain. Then, it uses a Laplacian in scale to select features which are invariant to scale.

An affine invariant detector is defined by Tuytelaars and Van Gool [337]. This starts from a local intensity maximum and searches along rays through that point to find local intensity extrema. The link, formed by those extrema, defines an interest region, which is later approximated by an ellipse. By searching along many rays and using ellipses to represent regions, the detected regions are invariant to affine transformation.

For the feature matching, the most robust descriptors are SIFT and its derivations. The SIFT descriptor [205] is a vector with 128 elements which is computed on the local image gradient. It uses a 4×4 regular grid around the feature and it computes for each grid the histogram of the image gradient. The eight bins values of each histogram become the values of the feature descriptor. SIFT is invariant to scale, rotation, illumination changes, noise and partially to view change.

Several improvements over SIFT have been proposed. In PCA-SIFT [174], Principal Component Analysis techniques are applied on the local patches of the image gradient to reduce the dimension of the descriptor; typically 36 elements. The result is a descriptor more robust to image deformation and more compact; this reduces the time for feature matching. In Gradient Location-Orientation Histogram (GLOH) [218], the descriptor is computed in a log-polar location grid around the feature and its size is reduced by PCA. In Cui et al. [76], the orientation histograms are computed on an irregular grid where patches are partially overlapped. This modification increases the robustness against the scale variation.

Another very efficient descriptor by Bay et al. [20] is called Speed Up Robust Features (SURF). SURF relies on the Haar wavelet responses around the feature and produces descriptors of 64 elements. The result is a descriptor as robust as the SIFT, but it reduces the time for features computation and matching.

While interest points are useful for finding image locations, that can be accurately matched in 2D, edge points are far more plentiful and often carry important semantic associations. For example, the boundaries of objects, which also correspond to occlusion events in 3D, are usually delineated by visible contours. Other kinds of edges correspond to shadow boundaries or crease edges, where surface orientation changes rapidly. Isolated edge points can also be grouped into longer curves or contours, as well as straight line segments.

From a qualitative point of view, edges occur at boundaries between regions of different color, intensity, or texture. Since segmenting an image into coherent regions is a difficult task, it is often preferable to detect edges using only local information, by defining for example an edge as a location of rapid intensity variation. Canny [60] discusses various filters and a number of researchers reviewed alternative edge detection algorithms and compared their performance [80, 232, 230, 89, 112, 231, 149, 75, 269, 38, 8].

2.1.2 Structure and motion

Structure and Motion is concerned with the recovery of the three dimensional geometry of the scene, the structure, when observed through a moving camera, the motion. Sensor data is either a video or an unstructured set of pictures; additional information, such as the calibration parameters, can be exploited if available. In other words, the problem of uncalibrated Structure and Motion from pictures is the problem of recovering a sparse three dimensional model of a scene given a set of images. One of the main applications of Structure and Motion has been the automatic reconstruction of architectural and urban scenarios. The sought result is generally a 3D point cloud and a set of camera matrices. The point cloud contains the interest points which were identified and tracked in the scene. The camera matrices identify the position and direction of each picture with respect to an arbitrary reference frame.

The main approaches to solve this problems are:

- real-time methods with constraints for urban environments [71, 258].
- batch methods without assumption on the scene and on the input images [48, 167, 170, 309, 340].

The first category currently rely on ancillary information, such as known camera calibration parameters, inertial navigation systems and GPS information. The second approach is more general and difficult, it has to cope with three challenges: generality, computational complexity and error accumulation. The generality issue refers to the assumptions on the lack of auxiliary information that is required in addition to pixels values. For the complexity issue, several different solutions have been explored. The most successful ones have been those aimed at reducing the impact of the *bundle adjustment* phase. This and the feature extraction phase dominate the computational complexity. The accumulation error plagues traditional structure and motion pipelines, and it can be tackled by hierarchical approaches that try to uniformly distribute errors.

Bundle adjustment is currently the most accurate way to refine the results of structure and motion estimation. It performs robust non-linear minimization of the re-projection errors. The term “bundle” refers to the bundles of rays connecting the 3D points with the camera centers, and the term “adjustment” refers to the iterative minimization of re-projection error. Bundle adjustment is also known in literature as “optimal motion estimation” [351] and as “non-linear least squares” [325, 320]. Triggs et al. [333] provide a good overview of this topic, also including pointers to the photogrammetry literature [305, 9, 184]. The topic is also treated in depth in textbooks and surveys on multi view geometry [100, 147, 220, 319].

To reduce bundle adjustment complexity, partitioning methods [107] have been proposed to subdivide the reconstruction problem into smaller and better conditioned subproblems which can be effectively optimized.

There are two main strategies to achieve this. The first strategy is to exploit regularities of the algorithm formulation in order to split the optimization problem into smaller components. Steedly et al. [315] apply spectral partitioning to structure and motion, thus selecting analytically the subproblems, while Ni et al. [239] exploit the underlying 3D structure of the problem for the subdivision. By this division strategy, the combinatorial complexity of the algorithms is kept under control when the number of images increases.

The other strategy works by limiting the image numbers growth and choosing subsets of them that encompass the entire solution. Fitzgibbon and Zisserman [107] use a balanced tree of trifocal tensors over a video sequence to perform efficient hierarchical sub-sampling. Nistér [240] refines the approach with heuristics to select tensor triplets and suppress redundant video frames, while Shum et al. [299] resolve different segments of the input sequence locally and then merge them hierarchically. This last approach has also been followed by Gibson et al. [126] by focusing on robustness aspects. These method have the advantage of improving the error distribution by distributing it on the whole dataset. Moreover, they are more robust to degenerate configurations. However, these solutions are specific for video streams and they cannot be trivially applied to typically unordered multi-view dataset.

Snavely et al. [310] propose a way to select a subset of the input image set whose reconstruction approximates the complete one. This approach removes redundancy, lowering considerably the computational requirements, but still process the images sequentially with the associated computational cost and error accumulation. Moreover, it still need to compute the epipolar geometry between all pairs of input images.

Agarwal et al. [2] propose a third kind of solution; orthogonal to the previous ones. This solution mitigates the problem by using more efficiently computational power, by subdividing reconstruction in small tasks and using load balancing techniques to improve timings. This approach encourages modifications to the dominant monolithic pipelines in order to optimize parallelization and workflow subdivision.

Existing pipelines either assume known internal parameters [48, 167], or constant internal parameters [170, 340], or rely on EXIF data plus external information (camera CCD size) [309]. While auto-calibration with varying parameters have been introduced several years ago [257]. The first working structure and motion pipeline that has been demonstrated with both varying parameters and no ancillary information is the recent SAMANTHA [97].

2.1.3 Multi-view stereo

The goal of multi-view stereo is to reconstruct a complete 3D object model from a collection of images taken from known camera view points and possibly from sparse 3D points from the output of Structure and Motion algorithms. The most challenging, but potentially most useful variant of multi-view stereo reconstruction, is to create globally consistent 3D models. This topic has a long history in computer vision, starting with surface mesh reconstruction techniques such as the one developed by Fua and Leclerc [114].

Techniques for producing 3D volumetric descriptions from binary silhouettes have been developed [261, 313, 318, 192], along with techniques based on tracking and reconstructing smooth occluding contours [67, 339, 371, 40, 321, 68].

A variety of approaches and representations have been used to solve this problem, including 3D voxel representations [288, 81, 186, 93, 304, 303, 345, 151], level sets [99, 260], polygonal meshes [233, 150, 120], and multiple depth maps [182].

Seitz et al. [287] developed a taxonomy to order and classify all those different approaches. The paper is paired with an evaluation website [286]. This website present results of comparisons and it has references to latest papers in the literature. The taxonomy divides the algorithms depending on their properties such as: the use of the scene representation, the photo-consistency measure, the visibility model, the shape priors, the reconstruction algorithm, and the initialization.

Between the various classes, one of the oldest approaches is to obtain the 3D surface by carving the volume of the object according to its silhouette in different views. In those *shape-from-silhouette* methods, the *visual hull* is defined as the maximal surface consistent with the object silhouette for all views. Most of these methods are based on segmentation, because images have to be segmented in background and foreground in order to delineate the object of interest with a consistent silhouette.

Recent approaches still integrate segmentations of images, but they are not limited to silhouette-based methods. For example, Yezzi and Soatto [367] employ a level set method, solved with a multi-resolution scheme, for exploiting the dual connection between the multi-view object segmentation and the 3D reconstruction of the underlying object. However, this system is not robust to the image noise and the initialization surface. This issue has beens solved by Kolev et al. [177] by reformulating the problem as a Bayesian estimation of the most probable shape that would yield to observed images.

In order to take into account the orientation of the evolving surface, Kolev et al. [180] generalize their previous approach by adding an anisotropy term into the energy optimization process. This energy minimization framework allows to combine multi-view photo-consistency, silhouette and normal information. Jancosek and Pajdla [169] also compute an over-segmentation of the dataset as a first step. However, this pre-processing is aimed to reduce computational load and providing priors for reconstructing flat areas of uniform colors. Campbell and Vogiatzis [57] compute segmentation using graph-cuts [42] on all images at the same time. They exploit the *fixation assumption*; all cameras points toward and they are centered on the objects that need to be segmented. A similar assumption is made in a previous work [59]. In this work, the graph-cut technique is applied on a voxel grid and images are used for silhouette coherency. In the work by Lee et al. [193], the background color is similarly extracted by intersecting all viewing volumes determined by cameras .

Sorman et al. [312] propose to cluster each image in the set using mean-shift, then clusters are segmented via graph-cut. Nevertheless, segmentation happens sequentially, whereupon each segmentation provides a shape prior to be used in the subsequent one.

Kolev et al. [179] deal with the image segmentation of all the views by projecting an evolving 3D surface. The problem is set as an energy minimization framework where the energy terms take into account a background and foreground terms plus a photo-consistency term. A variational solution is proposed by recasting the problem in a convex optimization one. A similar approach is used again by Kolev et al. [178], but this time three different energy terms are evaluated: a silhouette-based regional term with classic photo-consistency, a silhouette-based regional term with photo-consistency evaluated through a voting scheme, and a stereo-based regional constraint with denoised photo-consistency (the voting scheme).

Pons et al. [260] compute global matching scores on entire images from which projective distortion and semi-occluded regions have been removed. This avoids the complexities in matching windows of different shapes and tessellations of the tangent plane. For the computation of the matching score, only the pixels, that are visible from the position of the surfaces, are used.

Gargallo et al. [123] introduce the computation of the *exact* derivative of the reprojection error functional. This allows its rigorous minimization via gradient descent surface evolution. Delaunoy et al. [86] extends on this by employing a rigorous computation of the gradient of the reprojection error for non smooth surfaces defined by discrete triangular meshes. The gradient takes into account the visibility changes that occur when a surface moves. This forces the contours generated by the reconstructed surface to perfectly match with the apparent contour in the input images.

Multi-view stereo approaches share some similarities with video segmentation works [113, 347, 199]. However, approaches for videos can rely on coherence between frames and exploit optical flow, this is not possible with a multi-view dataset. The aforementioned methods usually do not work at depth map level, but they exploit volumetric representation or evolving surface of the model to reconstruct.

Many multi-view stereo algorithms work by estimating a depth map for each image and they then integrate such depth maps. Goesele et al. [129] propose a very simple algorithm to estimate the depth for each pixel by evaluating the photo-consistency. They use normalized cross-correlation of each estimated 3D point. The estimated depth is the one with the highest photo-consistency, and occlusions are taken into account by comparing against depth maps. Compared to other similar methods, this considers reliable only very high values of correlation. Finally, the generated sparse depth maps are merged together by applying the volumetric surface reconstruction algorithm of Curless and Levoy [77].

Bradley et al. [44] develop a very high quality method by proposing a viewpoint adaptive window to drive the stereo matching between image pairs. The dense high quality depth maps so generated are then merged together using a lower dimensional triangulation algorithm [134]. Recently, Xi and Duan [361] propose to integrate depth fusion algorithms; the graph-cut based global optimization is integrated with a mean-shift based explicit surface evolution.

Furukawa and Ponce introduce methods for multi-view stereo [116], these enforce both the photometric and geometric constraints associated with the input image data. Moreover, they exploit multi-view-stereo for bundle adjustment [119] by guiding the search for additional correspondences between the images using top-down information from rough estimates of camera parameters and the output of a multi-view-stereo system on scaled-down input images.

One of the most general and accurate algorithm for the 3D reconstruction from calibrated images is the one of Furakawa et al. [117]. This algorithm is the base of the well-known PMVS software [121], and it is based on a patch representation of the surface to reconstruct. The initial oriented patches are estimated, then expanded to the nearby pixels and filtered in an iterative way to obtaining a reliable and very dense reconstruction also in difficult cases.

With the availability of large collections of images through the Internet, many multi-view stereo algorithms started to exploit this kind of data. Goesele et al. [130] propose one of the first approaches of this kind, where the aim is to reconstruct objects from the Internet community photo collections. This data exhibits a tremendous variation in appearance and viewing parameters. To overcome this issue, compatible images need to be selected. Therefore, a global metric based on extracted SIFT features (similar lighting, weather, etc.) with wide-baseline from the set that are then matched is proposed. Finally, the surface is obtained by minimizing the re-projection error of oriented image patches with photogrammetry techniques.

An impressive example [2] of the maturity level of these technologies is the capability to reconstruct the most famous site of Rome in a day, starting from 2 millions of images on a cluster of PCs. Pollefeyes et al. [109] demonstrate the possibility to do the same without the availability of a large cloud of PCs. Furthermore, Furakawa et al. [117] propose a clusterization algorithm to allow his PMVS algorithm [118] to deal with huge unstructured image datasets.

An original approach in multi-view reconstruction has been proposed by Lambert and Hebért [190]. The main idea is to determine a *proxy surface* from a series of calibrated images without any assumption about the reflectance properties of the object to reconstruct. An initial estimation of the proxy surface is determined as the location of the *lumispheres*, that provide less variation, by minimizing the so-called frequency criterion. Since the lumisphere are built directly by the calibrated images, the reconstruction is obtained without perform any kind of matching. The obtained surface is then interpolated using Radial Basis Functions. Another interesting approach is the one of Campbell et al. [58], which employs multiple hypothesis to handle multiple depth values assigned to each pixel and exploits such information to obtain the reconstructed surface.

Web-based image mining MVS works are in general highlighted in modern literature, but there is a caveat: such approach actually does not work for many landmarks because people tend to take images from a very small variation of viewpoints, and less famous landmarks generate only few images with such search.

2.1.4 Appearance reconstruction

An important type of data, that should be added to three-dimensional models, is the surface appearance. This problem is related to several topics: controlled light environments, light modeling, material properties acquisition, and computational photography.

Unfortunately, the problem appears to be more complex than 3D reconstruction. For large artifacts, the setup for the acquisition of material properties using dedicated hardware is usually too complex for practical applications. Typically, the most used alternative approach is to rely on photographic information. These alternative techniques extract the color information or the reflectance field from a calibrated multi-view data set in order to re-synthesize the appearance of the surface. In this case, results are still heavily dependent on the quality of the starting dataset and the surface proprieties.

Essentially, if the light position is unknown, lighting artifacts (e.g. highlights, shadows, statically-fixed shading) are wrongly mapped on the 3D model as if they were color information. In dedicated acquisition campaigns, the photo acquisition is preferably performed under controlled lighting; in order to limit the presence of lighting artifacts. In this thesis, we focus on two of these subjects: color information acquisition and mapping, and illumination artifacts removal.

Color information acquisition and mapping

When mapping color information on 3D models is really difficult to acquire the complex material properties of a real object. An alternative solution is to try to obtain the "unshaded" color from a set of images. Firstly, the value is mapped on the digital object surface by registering those photos to the 3D model; computing the camera parameters. Secondly, the color is transferred from the images to the 3D surface by applying inverse projection. Despite the simple approach, there are several issues in selecting the correct color to be applied; especially when multiple candidates are present among different images. There is the need to deal with discontinuities, caused by color differences between photos that cover overlapping areas, and to reduce the illumination-related artifacts.

A first method to choose which color has to be applied to a particular area of the model is to select, for each part of the surface, an image following a particular criterion. In most of cases [56, 16, 195], this is the orthogonality between the surface and the view direction. In this way, only the "best" parts of the images are chosen and processed. Artifacts, caused by the discordance between overlapping images, are then visible on the border between surface areas, that receive color from different images. Between those adjacent images, there is a common redundant zone. This border can be used to obtain an adequate corrections in order to prevent sharp discontinuities.

This approach was followed by Callieri et al. [56], who propagate the correction on the texture space, and by Bannai et al. [16], who use the redundancy to perform a matrix-based color correction on the original image, and more recently by Gal et al. [122]. Other approaches, such as the one proposed by Lensch et al. [195], do not work only on the frontier area, but they blend on the 3D surface using the entire shared content to smooth out discontinuities.

Instead of cutting and pasting parts of the original images, as in previous approaches, a weight can be assigned to each input pixel. This value expresses the "quality" of its contribution. Then, the final color of the surface can be selected as the weighted mean of the input data using various quality metrics, as in Pulli et al. [263]. This weight-blend strategy has been introduced in several papers [25, 19, 265], with many variants in terms of number and nature of assembled metrics.

In particular, Callieri et al. [54] presented a flexible weighting system, that could be extended in order to accommodate additional metrics. A more recent work [88] uses flash light as a controlled light to enhance the color projection on 3D models.

Most of the analyzed methodologies present a common step: the possibility to *discard* parts of the input images or to selectively *assign a weight* to contributing pixels. These features can be extremely valuable once shadows have been detected: even if they are removed using image processing, the corrected portions can be assigned to a lower quality value in order to be used only if needed.

Illumination artifacts removal

The removal of artifacts from images is an operation which can be valuable for several fields of application, hence it has been widely studied. Several artifacts removal techniques have been proposed in the last few years. They can be roughly divided in two groups: the first one works on a single image [355, 323, 246, 294, 111, 102], which are mainly based on the analysis of the colors of the image. The second group uses a set of images [276, 200]; these algorithms exploit the redundancy between images. In general, all these methods assume no prior information about the geometry of a scene.

More recently, the use of flash/no-flash pairs to enhance the appearance of photographs has been proposed in several interesting papers. These works [154, 92, 251] propose techniques to enhance details and to reduce noise in ambient images. Furthermore, they present straightforward ways to remove shadows and highlights. Although results are very interesting, these techniques can be applied only when the flash light is the dominant one in the scene. This limits their use in outdoor environments. Flash/no-flash pairs can detect and remove ambient shadows [207]. In the work by Dellepiane et al. [88], a framework for the detection and removal of lighting artifacts produced by flash light is presented.

2.2 Related work on image localization and registration

Image localization is also a vast field. Here, we present a brief overview of some of the most relevant publications in chronological order.

Morris and Smelyanskiy [223] face the problem of single image calibration over a 3D surface and the simultaneous surface refinement based on additional information given by the image. The algorithm is based on the extraction of image salient points, i.e. Harris detector [146], and it employs minimization of an objective function via gradient calculation. The approach works relatively well only when a good initial estimate of the surface is provided, and it is not scalable.

Schaffalitzky and Zisserman [279] build a sort order for sets of photographs through a calculation of correspondences between pairs of photographs. The method is relatively robust with respect to the positional distance photographs. This approach allows for the "stitching" of the various photographs, assuming that there is enough overlap between the images. GPS-based coordinate triangulation is not performed. This paper has been the basis of many commonly-used image-based 3D reconstruction techniques, such as PhotoTourism and Photosynth [309].

Shao et al. [292] treat the problem of database-based image recognition by comparing them to a reference image through the use of local salient features. These are described independently of possible affine transformations between them.

Wang et al. [348] propose a solution for the Simultaneous Localization And Tracking (SLAM) robotic problem [307]. A database of salient points, extracted from the robot camera, is used for the localization. SLAM approaches suffer from the "Kidnap problem", i.e. the inability to continue the localization and mapping between non-contiguous locations.

Cipolla et al. [69] try to solve this by applying wide baseline matching algorithm techniques between a digital photo and a geo-referenced database. The main limitation of this approach comes from the manual construction of the database correspondences between the map and the photos. Robertson and Cipolla [271] propose an improvement of it by exploiting the perspective lines relative to the vertical edges of buildings.

Zhang and Kosecka [370] built a prototype for urban localization of images that relies on a photographic database augmented with GPS information. The system extracts one or more reference images and from these localized the input image.

Paletta et al. [247] define a specific system devoted to the improvement in the description of the images' salient points, called "informative-SIFT".

Gordon and Lowe [135] propose the first work which exploits Structure and Motion for precise localization of the input image. This approach provided interesting ideas in later works [168, 198, 278].

Schindler et al. [283] face the problem of localization in very large datasets of streets' photographs using a tree data structure that indexes the salient features for scalability.

Zhu et al. [374] built another system for large-scale global localization, with very high accuracy due to the use of 4 cameras, arranged as two stereo pairs.

Xiao et al. [362] propose a method for the recognition and localization of generic objects from uncalibrated images. The system includes an interesting algorithm for simultaneous localization of objects and camera positions. This combines segmentation techniques, example models and voting techniques. The main purpose of the system is object recognition using structural representation in 3D space.

Irschara et al. [168] propose a localization system that effectively exploits image-based 3D reconstruction. After the reconstruction, each 3D point is associated to a compressed description of the features of the images incidents therein. Such descriptions are indexed using a tree-based vocabularies for efficient searching.

Li et al. [198] propose another feature-based approach based on a prioritization scheme. The priority of a point is related to the number of cameras from the reconstruction it is visible in. The use of a reduced set of points of highest priority has several advantages with respect to using all 3D points. This method, in terms of the number of images that can be registered, outperforms the algorithm by Irschara et al.

Recently, Sattler et al. [278] proposed a direct 2D-to-3D matching framework. By associating 3D points to visual words, they quickly identify possible correspondences for 2D features which are then verified in a linear search. The final 2D-to-3D correspondences are then used to localize the image using N-point pose estimation.

Our proposal, presented in Chapter 5, follows ideas similar to the methods of Irschara et al. and Sattler et al.; however, it can provide more accurate and reliable results thanks also to a novel validation phase.

2.3 Related work on visualization of multi-view data

Image-Based Rendering (IBR) has emerged in the last 20 years as one of the most novel applications in-between Computer Graphics and Computer Vision [173, 297]. Graphics rendering techniques are combined with 3D reconstruction techniques, using multi-view image data to create interactive and photo-realistic experiences.

The many IBR algorithms developed in these years are usually categorized according to the amount and quality of the geometry employed to generate the final image [194]. For example, *billboarding* [3], which uses a very minimal amount of geometry, i.e. an oriented quadrilateral, with a texture applied to create an impostor of the object to render (numerous declinations of this technique exists). *View interpolation techniques* which relies on more geometric data to work as Chen et al. [62] that creates a seamless transition between a pair of reference images using one or more pre-computed depth maps, layered depth images [289]. This attempts to improve the use of sprites and depth. Some other techniques, such as *view-dependent texture maps* [84], exploit a proxy of the model geometry to project and blend multiple texture maps on the 3D surface. Another technique that is worthwhile to mention is the Lumigraph [136], which employs a large amount of images and does not rely on any geometric information to build a representation of a *light field* through a four-dimensional parameterization of it. This allows to render a scene from any arbitrary viewpoint. Many variants of this idea have been proposed such as environment mattes [375], concentric mosaics [298], surface light fields [357], and recent progresses such as unstructured Lumigraph [50] and the unstructured light fields [79].

Recently, the diffusion of multi-view image sets has powered the development of new type of browsers for the visualization of this type of data. Multi-view data sets can be visualized as simple image collections, or by hybrid 2D/3D visualization systems able to exploit their inherent 2D/3D characteristics. The rendering of those kinds of data can be declined in various ways, depending both on the 3D model capturing/reconstruction approach and the desired visual effects. 3D models are usually represented as triangulated meshes, or as point clouds, and calibrated photographs can be used to carry out the necessary color information to complement or substitute the appearance reconstruction step. This topic focus in particular on how to enhance the user experience during the navigation. In the following, we put our attention on these types of work, that are closely related to the visualization system proposed in Chapter 6.

2.3.1 Effective browsing of large image datasets

Traditionally, basic image browsers just display all thumbnails over a panel, which can be scrolled to navigate the dataset [133]. In presence of larger datasets, this approach becomes quickly unfeasible, and hierarchical, focus-and-context browsing mechanisms become necessary. The hierarchy can be based on information extracted form the pictures or coming from other sources, like geo-tagging or any other metadata.

In PhotoMesa [21] the hierarchy reflects the file system organization, while PhotoTOC [252] proposes a one-level clustering algorithm, based on color analysis and time of shot, to select an overview of the images and show the detailed view of one cluster per time.

Further work has been conduced to show more intuitive user interfaces, exploiting time [164] and color [275] information. Other browsers also exploit the 3D localization of the photographs, mainly exploiting GPS EXIF tags [226].

Recent advances in Structure and Motion have provided the basis for geo-referenced clusterings [94], which can sum up large datasets in a few general images, provided that such images are present in the dataset. A self-reorganizing Voronoi-based approach is used in Brivio et al. [46] to arrange thumbnail positions, clustering, and sizes in a focus-and-context way according to any of the above measures.

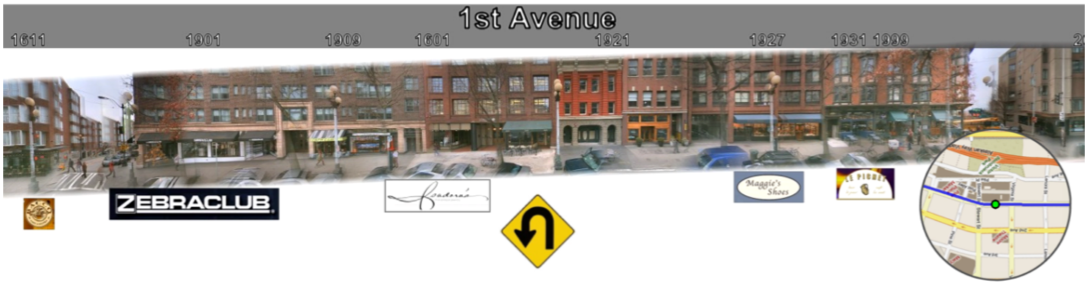


Figure 2.1: A multi-perspective street slide panorama with navigational aides and mini-map (from Kopf et al. [183]).

2.3.2 Integration of 2D and 3D datasets

Recently, some effort have been put in build visualization systems aiming at visualizing directly the multi-view image dataset with in some way. This leads to the development of some browsers supporting the simultaneous navigation of mixed 2D and 3D datasets [309, 341, 308, 183, 128], each one tackling different technical goals. Pioneer work on virtual exploration dates back to the late 1970s. Through an interactive projection approach of photographs, the Movie-Maps project [202] focused on virtual travel along the streets and façades of the main buildings of the city of Aspen. Photographs were acquired with an instrumented rig of four cameras mounted on a truck, shooting once every 10 traveled feet. At runtime, a user could interactively choose his walk across the city photographs, and could select some buildings for a more detailed view and optionally some video of their insides. The system also offered the possibility to switch between two seasonal datasets and integrated different data sources, including textual and acoustic.

The Movie-Maps image acquisition system was also a precursor of Google Street View [341]. The latter takes advantage of more recent devices such as accelerometers and GPS to feed its Structure from Motion system. Then, the output calibration enables blending among overlapping images, whereas the sparse point cloud is used to infer approximate planar structures position, size, and orientation. During the navigation, the scene is divided into *bubbles/panoramas*, each composed of the nearest photographs visible by means of rotations around the current view position. Street View is currently available as a streamed application, which needs only the current bubble to be downloaded. More recently, Street Slide [183] introduces a technique to smoothly switch from panorama bubbles to a multi-perspective view, and vice-versa. This gives a broader planar view of streets, see Fig. 2.1. The system is explicitly designed to exploit street features, thus also displays additional information, such as street names and shop banners.

Photo Tourism [309] is the first proposal for an explicit integration and rendering of a sparse 3D geometry and a photo collection (see Fig. 2.2). It allows the user to navigate through a large collections of images by exploiting the underlining

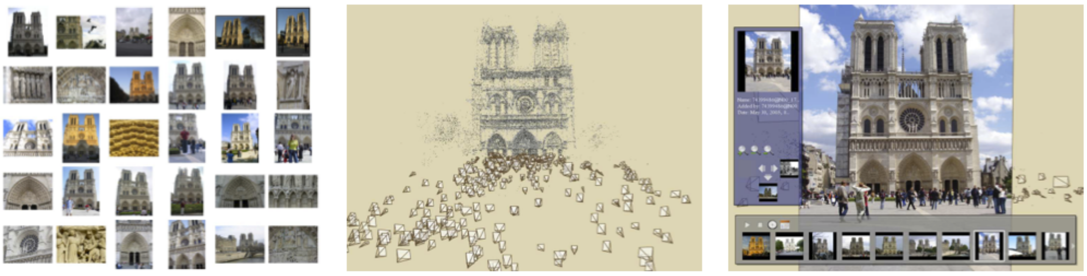


Figure 2.2: Photo Tourism [309], a system for interactively browsing and exploring large unstructured collection of photographs.

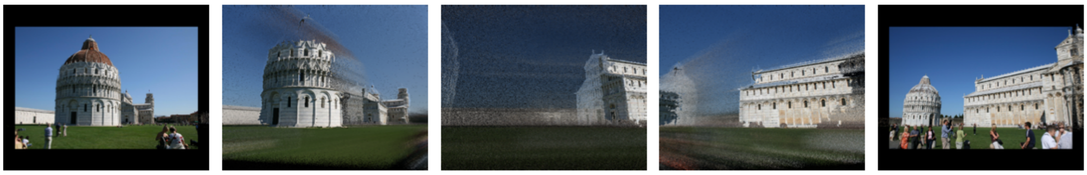


Figure 2.3: An extreme view transition breaking the limits of traditional view-based rendering. The global and ambient point clouds fill unknown regions providing a good 3D impression. Foreground occluders dissolve smoothly during the transition. (from Goesele et al. [128]).

3D structure relationships obtained through a Structure and Motion algorithm. The central region of its interface is devoted to render the 3D scene overlaid with the currently selected photograph. Further details about the photograph are rendered in a side menu, including the thumbnails of those photographs overlapping it at its sides. A thumbnail-bar contains photographs depicting the current subject from different views. Among its features, nearby photographs can be browsed, optionally viewing them as a slideshow, and to jump to remote dataset areas through an overhead map. Its current commercial version is the widely popular Photosynth [216].

Since the presentation of Photo Tourism, other visualization systems have been proposed sharing the same input, each tackling a different sub-problem. Photo Tourism's navigation strategy, driven by photograph selections, is then improved in Snavely et al. [308] by providing navigation controls which help the user to move along paths of dense photographs. These include orbit, panorama, and path planning between two arbitrary photographs. The underlying mechanism is based on scores assigned to photographs, which encode how well an object of interest is viewed from each image. Finally, a very recent work introduces the Ambient Point Clouds [128] to approximate view interpolations along those view rays whose projections correspond to uncertain or incomplete geometry, see Fig. 2.3. In that system, depth maps are used to render geometry within a negligible error from the view of the current active photograph. The rest of the scene is represented by a subsampled point cloud for efficiency reasons.

Chapter 3

Multi-view processing for image-based 3D reconstruction

*On the side of **multi-view processing for image-based 3D reconstruction**, we present an image pre-processing algorithm, that is a special color-to-gray conversion, that improves the accuracy of general image-based reconstruction algorithms by means of a* domain separation *strategy. We explore the space of image pre-processing techniques in order to optimize the input for general classes of 3D reconstruction techniques, with the aim to understand the conversion qualities that can improve the accuracy of results when the grayscale conversion is applied as a pre-processing step in the context of vision algorithms, and in particular dense stereo matching. To achieve this, we evaluate many different state of the art color to grayscale conversion algorithms. We also propose an ad-hoc adaptation of the most theoretically promising algorithm, which we call Multi-Image Decolorize (MID). This algorithm comes from an in-depth analysis of the existing conversion solutions and consists of a multi-image extension of the algorithm by Grundland and Dodgson [138] which is based on predominant component analysis. In addition, two variants of this algorithm have been proposed and analyzed: one with standard unsharp masking and another with a chromatic weighted unsharp masking technique [241] which enhances the local contrast as shown in the approach by Smith et al. [306]. We test and show results in the case of dense stereo matching algorithm, showing how matching results can be enhanced in different algorithms by pre-computing a single optimized numerical value for each image location. We tested the relative performances of this conversion with respect to many other solutions, using the StereoMatcher test suite [282] with a variety of different datasets and different dense stereo matching algorithms. The results show that the overall performance of the proposed MID conversion is good. Moreover, the reported tests provided useful information and insights on how to design color to gray conversion for improving matching performance. We also show some interesting secondary results such as the role of standard unsharp masking vs. chromatic unsharp masking in improving correspondence matching.*

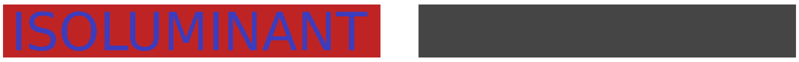


Figure 3.1: Isoluminant changes are not preserved with traditional color to grayscale conversion. Converting a blueish text whose luminance matches that of the red background to grayscale can result in a featureless image.

3.1 Introduction

This Chapter tackles the color to grayscale conversion of images. The main goal of this advancement is to understand what can improve the quality and the accuracy of results when the grayscale conversion is applied as a pre-processing step in the context of stereo and multi-view stereo matching. We evaluated many different state-of-the-art algorithms for color to gray conversion and also attempted to adapt the most promising algorithm (from a theoretical viewpoint). This has lead to the creation of an ad-hoc algorithm that optimizes the conversion process by simultaneously evaluating the whole set of images.

Color to grayscale conversion can be seen as a *dimensionality reduction* problem. This operation should not be undervalued, because there are many different properties that need to be preserved. For example, as shown in Fig. 3.1, isoluminant color changes are usually not preserved with commonly used color to gray conversions. Many conversion methods have been proposed in recent years, but they mainly focus on the reproduction of color images with grayscale mediums. Perceptual accuracy in terms of the fidelity of the converted image is often the only objective of these techniques. These kinds of approaches, however, are not designed to fulfill the needs of vision and stereo matching algorithms.

The problem of the automatic reconstruction of three-dimensional objects and environments from sets of two or more photographic images is widely studied in Computer Vision [147]: traditional methods are based on matching features from sets of two or more input images, as explained in Sec. 2.1. While some approaches [282] use color information, only a few solutions are able to take real advantage of the color information. Many of these reconstruction methods are conceptually designed to work on grayscale images in the sense that, sooner or later in the processing, for a given spatial location, the algorithm will only consider a single numerical value (instead of the RGB triple). Often, this single numerical value is the result of a simple aggregation of color values.

While finding an optimal way to exploit complete RGB information in stereo matching would be interesting, we preferred to focus on the color to gray conversion. A better exploitation of color information during the matching would need to be implemented for each available matching algorithm in order to maximize its usefulness and to assess its soundness. In contrast, working on an enhanced color to gray conversion step could straightforwardly improve the performance of an entire class of existing and already well-known reconstruction algorithms. In other words, we followed a *domain separation* strategy, since we decouple the color treatment from the computer vision algorithm using a separate preprocessing step for the aggregation of the data.

The aims of this work are twofold. Firstly, to provide a wide and accurate comparison of the performance of existing grayscale techniques. Secondly, to develop a new conversion technique based on the existing one by analyzing the needs of matching algorithms.

In general, three approaches can be used to evaluate the correctness of different color to grayscale conversion algorithms:

- A perceptual evaluation, such as the one employed in Čadík's 2008 article [53], is best suited for grayscale printer reproduction and other human-related tasks.

- An information theory approach could quantify the amount of information that is lost during the dimensionality reduction; to the best of our knowledge there are no other similar studies in this context.

- An approach that is tailored to measure the results of the subsequent image processing algorithms.

We use the third approach, by evaluating the *effectiveness* of different grayscale conversions with respect to the image-based reconstruction problem. We chose a well-known class of automatic reconstruction algorithms, i.e., *dense stereo matching* [282] and we tested the performance of the traditional color approach compared to many different conversion algorithms. In dense stereo matching, in order to compute 3D reconstructions, the correspondence problem must first be solved for every pixel of the two input images. The simplest case occurs when these images have been rectified in a fronto-parallel position with respect to the object. A dense matching algorithm can compute a map of the horizontal disparity between the images that is inversely proportional to the distance of every pixel from the camera. Given two rectified images, these algorithms perform a matching cost computation. Then, they aggregate these costs and use them to compute the disparity between the pixels of the images.

We separate the color treatment from the matching cost computation using a preprocessing step for the grayscale conversion and we compared the results between different conversions of the same datasets. This approach allows us to assess the pitfalls and particular needs of this field of application.

Our conversion is based on an analysis of both performances and characteristics of the previously selected algorithms, and it optimizes the process by simultaneously evaluating the whole set of images that needs to be matched. Two variants of the base technique that recover the local loss of chrominance contrast are also proposed and tested.

3.1.1 Contributions

The contributions of this work can be summarized as:

- An analysis of the characteristics of many different state of the art color to gray conversion algorithms in the context of stereo matching.

- A comparison of the performances of these algorithms in the context of dense stereo matching.

- Thanks to the wide range of techniques evaluated and the level of detail of their respective descriptions, the next background section can be seen as a general survey on color to gray conversion techniques.

- Multi-Image Decolorize (MID), an ad-hoc grayscale method based on a theoretical analysis of the requirements and characteristics of existing methods. This technique can be considered as a first attempt to design a grayscale conversion specific for the task of dense and multi-view stereo matching.

3.2 Background

In this section, we give a detailed overview of color to gray conversion algorithms; also considering issues in gamma compression. Then, we describe the role of color information in stereo matching.

3.2.1 Color to gray conversions

Colors in an image may be converted to shades of gray by calculating, for example, the effective brightness or luminance of the color and using this value to create a shade of gray. This may be useful for aesthetic purposes, for printing without colors and for image computations that need (or can be speeded up using) a single intensity value for every pixel. Color to grayscale conversion performs a reduction of the three dimensional color data into a single dimension.

A standard linear technique for dimensionality reduction is Principal Component Analysis (PCA). However, as explained in [266], PCA is not a good technique for color to gray conversion due to the statistical color properties commonly found in input images. This kind of color clustering undermines the efficiency of the PCA approach by underexploiting the middle-range of the gamut.

Moreover, some information, during the conversion, is lost due to the PCA nature. Therefore, the goal is to save as much information from the original color image as possible. Hereafter, we use *information* to refer to the information used to produce “the best” grayscale results for a specific task. For example, the best conversion may be the most perceptually accurate (i.e., the converted image is perceptually similar to the original even if color is discarded) or the one that maximizes some specific global properties such as luminance or contrast.

Many different color spaces [26, 95, 268, 293] are used for color to grayscale conversions and over the last few years many advanced approaches to this problem have been proposed [14, 132, 138, 139, 238, 82, 266, 306]. Color to gray conversions can be classified into two main categories: *functional* and *optimizing*. Functional conversions are image-independent *local* functions of every color, e.g., for every pixel of the color image a grayscale value is computed using a function whose only parameters are the values of the corresponding color pixel. Optimizing conversions are more advanced techniques which depend on the whole image that needs to be converted. They can use spatial information and global parameters to estimate the best mapping and to preserve certain aspects of the color information.

Functional Grayscale conversions

Functional conversions can be subdivided into three subfamilies: *trivial methods*, *direct methods* and *chrominance direct methods*. Trivial methods do not take into account the power distribution of the color channels; for example, only the mean of the RGB channels is taken. Informally speaking, they lose a lot of image information, because for every pixel they discard two of the three color values, or discard one value averaging the remaining ones, not taking into account color properties. Direct methods are standard methods where the conversion is a linear function of the pixel's color values, good enough for non-specialized uses. Typically, this class of functions takes into account the spectrum of different colors. These first two categories are widely used by many existing image processing systems. Chrominance direct methods are based on more advanced color spaces and can mitigate the issue when having isoluminant colors.

Trivial methods Trivial methods are the most basic and simple ones. Despite the loss of information, these color to grayscale conversions are commonly used for their simplicity. We briefly describe four of the most common methods in this class, roughly sorted from worst to best in terms of the (approximate) preservation of information.

The *Value HSV* method takes the HSV representation of the image and uses Value V as the grayscale value. This is equivalent to choosing for every pixel the maximum color value and using it as the grayscale value. This method loses the information relative to which color value is kept for a pixel. Another problem is that the resulting image luminance is heavily biased toward white.

The *RGB Channel Filter* selects a channel between R, G or B and uses this channel as the grayscale value. The green filter gives the best results and the blue filter gives the worst results in terms of lightness resemblance. In this case, the color transformation is consistent for all the pixels in the image.

Lightness HSL: takes the *HSL* representation of the image and uses Lightness L as the grayscale value. This value is the mean between the maximum and the minimum of the color values. In this method, a color value is discarded from every pixel, the remaining values are averaged and the information is lost in terms of which color value is discarded for a pixel.

The *Naive Mean* takes the mean of the color channels. The advantage of this method, compared to the other trivial ones, is that it takes information from every channel, though it does not consider the relative spectral power distribution of the RGB channels.

Direct methods An easy improvement over trivial methods is to calculate the grayscale value using a weighted sum over the color channels. Using different weights for different colors means that factors such as the relative spectral distribution of the color channels and the human perception can be taken into account. Many of the most used grayscale conversion are based on a method of this family. We describe three of the most representative of these methods.

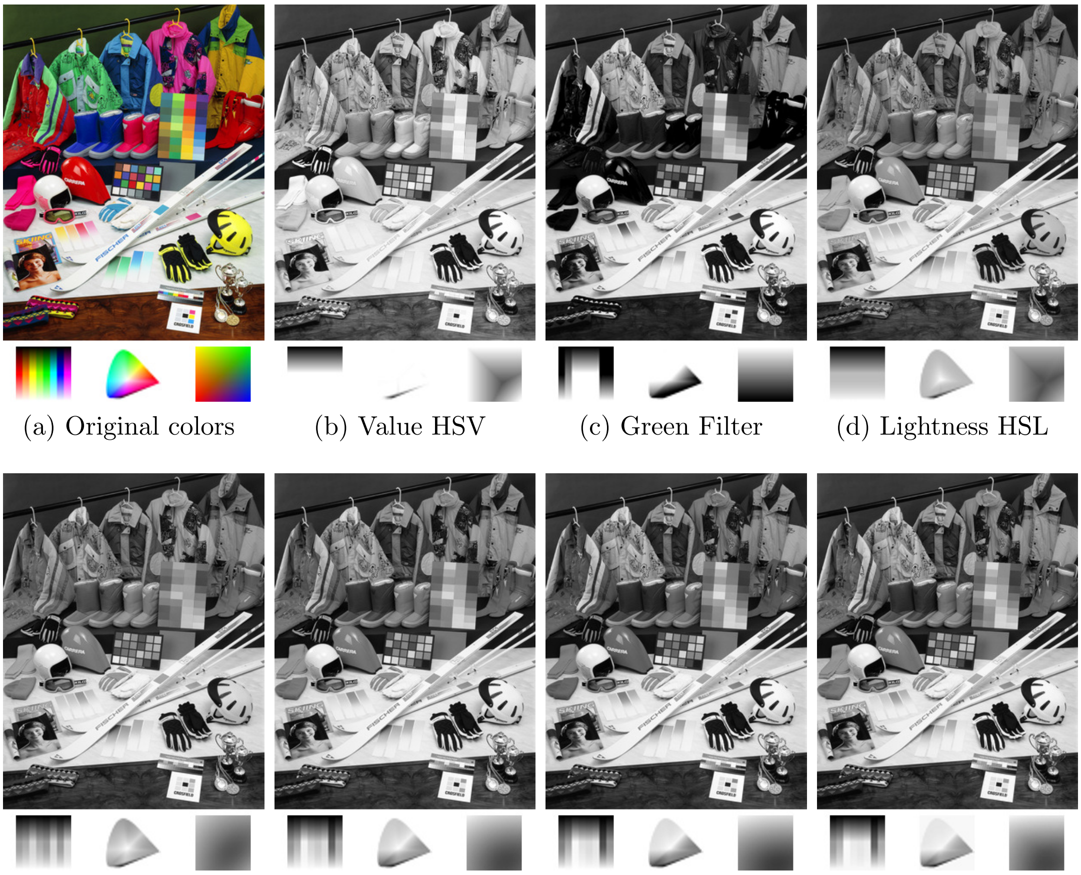
The *CIE Y* method is a widely used conversion that is based on the *CIE 1931 XYZ* color space [142, 359]. It takes the XYZ representation of the image and uses Y as the grayscale value.

The *NTSC* method is another widely used conversion (NTSC Rec.601) created in 1982 by the ITU-R organization for *luma* definition in gamma precompensated television signals.

The *QT builtin* method is an example of a grayscale conversion using integer arithmetic. It is an approximation of the NTSC Rec.601 (implemented in the qGray function of Trolltech's QT framework) and is designed to work with integer representation in the $[0..255]$ range.

Chrominance direct methods A problem with the above approaches is that the distinction between two different colors of similar "luminance" (independently of its definition) is lost. *Chrominance direct methods* are based on more advanced considerations of color spaces compared to the previous ones, and have been defined specifically to mitigate this problem. These conversions are still local functions of the image pixels, but they assign different grayscale values to *isoluminant* colors. To achieve this result, the luminance information is slightly altered using the chrominance information. In order to increase or decrease the "correct" luminance to differentiate isoluminant colors, these methods exploit a result from studies on human color perception: the *Helmholtz-Kohlrausch (H-K) effect* [96, 95, 306]. The H-K effect states that the perceived lightness of a stimulus changes as a function of the chroma This phenomenon is predicted by a chromatic lightness term that corrects the luminance based on the color's chromatic component and on a starting color space. We examined three of such predictors.

The *Fairchild Lightness* [96] method is a chromatic lightness metric that fits to



(a) Original colors (b) Value HSV (c) Green Filter (d) Lightness HSL

(e) Naive Mean (f) NTSC Rec.601 (g) CIE Y (h) Nayatani VAC

Figure 3.2: An example of some Functional grayscale conversions

data [360] using a cylindrical representation of the CIE L*a*b* color space called *CIE L*a*b* LCH*; lightness, chroma and hue angle.

The *Lightness Nayatani VAC* [235, 236, 237] method is based on a chromatic lightness metric defined on the *CIE L*u*v** color space and the Variable-Achromatic-Color (VAC) approach, in which an achromatic sample's luminance is adjusted to match a color stimulus. VAC was used in the 1954 Sanders-Wyszecki study and in Wyszecki's 1964 and 1967 studies [360].

The *Lightness Nayatani VCC* method is based on another chromatic lightness metric defined by Nayatani [236]. It is based on the *CIE L*u*v** color space and the Variable-Chromatic-Color (VCC) approach, in which the chromatic content of a color stimulus is adjusted until its brightness matches a given gray stimulus.

VCC is less common than VAC and its chromatic object lightness equation is almost identical to the VAC case[1]. A quantitative difference between them is that VCC lightness is twice as strong as VAC lightness (in log space). Moreover, it has been observed [236, 306] in VCC lightness that its stronger effect maps many bright colors to white. This makes impossible to distinguish between very bright isoluminant colors. For a much more detailed description of these metrics and a clear explanation of their subtle differences see Nayatani's 2008 paper [237].

As Fig. 3.2 shows, the first three conversions ((b), (c) and (d)) discard a lot of information (observe the color swatches) and lose features, thus affecting perceptual accuracy and also potential matching. Channel averaging (e) gives “acceptable” results at least for human perception. There are not many noticeable differences between the last three cases ((f), (g) and (h)).

Optimizing Grayscale conversions

We present eight advanced techniques that constitute the state of the art in this field. For the sake of simplicity, we name these methods using the surname of the first author and a mnemonic adjective taken from the title of the relative paper. Some of these conversions can be roughly aggregated in the categories described in the following.

Three perform a functional conversion and then optimize the image using spatial information in order to recover some of the characteristics that have been lost:

- the *Bala Spatial* [14] method adds high frequency chromatic information to the luminance.
- the *Alsam Sharpening* [4] method combines global and local conversions.
- the *Smith Apparent* [306] method, similar to the Alsam Sharpening method.

Two methods employ iterative energy minimization:

[1] See Section 3.3.2 for the mathematical definition of VAC, the VCC equation differs only by having a constant set to -0.8660 instead of -0.1340.

- the *Gooch Color2Gray* [132] method finds gray values that best match the original color differences through an objective function minimization process.
- the *Rasche Monochromats* [266] method tries to preserve image detail by maintaining distance *ratios* during the dimensionality reduction.

Finally, there are other orthogonal approaches that do not closely fit with the previous classes:

- The *Grundland Decolorize* [138, 139] method finds a continuous global mapping which tries to put back the lost chromatic information into the luminance channel.
- The *Neumann Adaptive* [238] is heavily based on perceptual experimental measures. More specifically, the method stresses perceptual loyalty by measuring the image's gradient field by color differences in the proposed Coloroid color space.
- The *Queiroz Invertible* [82] exploits the wavelet theory in order to hide the color information in "invisible" bands of the generated grayscale image. This information encoded into the high frequency regions of the converted image can be later decoded back to recover part of the original color.

We briefly explain these techniques roughly in chronological order. In Section 3.3, we give further details about the conversions used in our tests.

Bala Spatial In their work on the study of chromatic contrast for grayscale conversion, Bala et al. [14] take a spatial approach and introduce color contrasts in the CIE L*a*b* LCH cylindrical color space by adding a high-pass filtered chroma channel to the lightness channel. More intuitively, they enhance the grayscale image with the contours of the chromatic part of the image. To prevent overshooting in already bright areas, this correction signal is locally adjusted. The algorithm is susceptible to issues in chroma and lightness misalignment.

Alsam Sharpening Alsam and Kolås [4] introduced a conversion method that aims to create sharp grayscale from the original color rather than enhancing the separation between colors. The approach resembles the Bala Spatial method: firstly, a grayscale image is created by a global mapping to the image-dependent gray axis. Secondly, the grayscale image is enhanced by a correction mask in a similar way to unsharp masking [131].

Smith Apparent A recent method by Smith et al. [306] combines global and local conversions in a similar way to the Alsam Sharpening method. The algorithm applies a global "absolute" mapping based on the Helmoltz-Kohlrausch effect, and then locally enhances chrominance edges using adaptively-weighted multi-scale unsharp masking [241]. While global mapping is image independent, local enhancement reintroduces lost discontinuities only in regions that insufficiently represent the original chromatic contrast [306]. The main goal of the method is to achieve perceptual accuracy without exaggerating the features discriminability.

Gooch Color2Gray Gooch et al. [132], introduced a local algorithm known as *Color2Gray*. In this gradient-domain method, the gray value of each pixel is iteratively adjusted to minimize an objective function, which is based on local contrasts between all pixel pairs. The original contrast between each pixel and its neighbors is measured by a signed distance, whose magnitude accounts for luminance and chroma differences and whose sign represents the hue shift with respect to a user defined hue angle.

Rasche Monochromats Rasche et al.'s method [266] aims to preserve contrast while maintaining consistent luminance. The authors defined an error function based on matching the gray differences to the corresponding color differences. The goal is to minimize the error function for finding an optimal conversion. Color quantization is proposed to reduce the considerable computational cost of the error minimization procedure.

Grundland and Dodgson [138, 139] performed a global grayscale conversion by expressing grayscale as a continuous, image-dependent, piecewise linear mapping of the primary RGB colors and their saturation. Their algorithm, called *Decolorize*, works in the YPQ color opponent space. In this color space, the color differences are projected onto the two *predominant* chromatic contrast axes and are then added to the luminance image. Unlike the principal component analysis, which optimizes the variability of observations, the predominant component analysis optimizes the differences between observations. The predominant chromatic axis aims to capture, with a single chromatic coordinate, the color contrast information that is lost in the luminance channel. Since this algorithm constitutes the main basis of the ad-hoc adaptation Multi-Image Decolorize, a detailed description is given in Section 3.3.5. The Multi-Image Decolorize is described in Section 3.4.

Neumann Adaptive Neumann et al. [238] presented a local gradient-based technique with linear complexity that requires no user intervention. It aims to obtain the best *perceptual* gray gradient equivalent by exploiting their Coloroid perceptual color space and its experimental background. The gradient field is corrected using a gradient inconsistency correction method. Finally, a 2D integration yields the grayscale image. In the same paper they also introduce another technique which is a generalization of the CIE L*a*b* formula [95]. This can be used as an alternative to the Coloroid gray gradient field.

Queiroz Invertible Queiroz and Braun [82] have proposed an invertible conversion to grayscale. The idea is to transform colors into high frequency textures which are applied onto the gray image and can be later decoded back to color. The method is based on wavelet transformations and on the replacement of sub-bands by chrominance planes.

A note about gamma compression and grayscale conversions

Gamma correction is a nonlinear operation used to compress or expand luminance or tristimulus values in video or still image systems. All image processing algorithms should take into account such gamma precompensation in order to be properly applied. The main problem is that, image's gamma is often unknown. Moreover, many applications/algorithms ignore this issue. For these reasons, it is interesting to discuss how the grayscale conversions considered so far are influenced by the knowledge of the image's gamma.

With regard to the naive methods, Value HSV and RGB channel filters are not at all affected by the gamma, since they do not manipulate color values but only choose one of them. The other functional techniques are relatively robust from this point of view, although applying these conversions to gamma precompensated values is not theoretically sound. The impact of this issue for advanced techniques is difficult to predict; although from practical experience, Bala Spatial, Alsam Sharpening and Smith Apparent would seem to be the most robust. This is because they are basically a weighting of color values with the spatial driven perturbations that enhance them. A study of the effects of this issue in approaches such as Gooch Color2Gray, Rasche Monochromats, Neumann Adaptive and Queiroz Invertible would be very complex and is out of the scope of this work.

We underline that Grundland Decolorize and, consequently, our Multi-Image Decolorize technique are both very sensible to this issue, because they use saturation and the proportions between the image chromaticities to choose the mapping of a color hue to increases or decreases in the basic lightness. If the values are not linear, these ratios change significantly and the resulting mapping is very different. We come back to this point in Section 3.3.5.

3.2.2 Color and grayscale in matching

Few articles deal with color based matching. The simplest approaches take the mean of the three color components or aggregate the information obtained from the single channels in some empirical way. Of the few studies on the correlations between color and grayscale in matching algorithms, we can cite Chambon and Crouzil [61] and Bleyer et al. [35] works.

Chambon and Crouzil [61] propose an evaluation protocol that helps choosing a color space and to generalize the correlation measures to color. They investigated nine color spaces and three different methods of computing the correlation needed in the matching cost computation phase of stereo matching in order to evaluate their respective effectiveness.

Bleyer et al. [35] continue Chambon and Crouzil's work by inspecting the effects of the same color spaces and methods in the specific field of global dense stereo matching algorithms which optimizes an energy function via graph-cuts or belief propagation.

Compared with color stereo matching, our domain separation approach has several advantages. Firstly, the computational time required for the overall processing can be less computationally expensive. Secondly, it can be applied to different stereo matching algorithms, because it is a pre-processing step. Thirdly, in the experimental results (see Section 3.5), we show that, in some cases, a proper color to gray conversion could give *better* results than color processing. Finally, the potential benefits could probably be also employed in other scenarios such as the generation of more robust local features [336] in sparse matching and the improvement of multi view stereo matching algorithms [340].

3.3 Details about the tested conversions

When choosing the algorithms to test in the stereo matching context, we wanted to cover a wide range of approaches. Concerning functional conversions, we chose the CIE Y direct method as a general representative and the Lightness Nayatani VAC because of its relationship with the Smith Apparent technique. Among the eight optimizing techniques described in Section 3.2.1, we selected and implemented Gooch Color2Gray, Smith Apparent and Grudland Decolorize for the following reasons:

- Queiroz Invertible was discarded because its aim is to hide color information and not to preserve details in the converted image. Therefore, it does not improve feature discriminability with respect to classical conversions.

- Rasche Monochromats and Neumann Adaptive were not considered due to the color quantization issue and the unpredictable behavior in inconsistent regions of the gradient field.

- Of three similar techniques, Bala Spatial, Alsam Sharpening and Smith Apparent, we decided to test the most recent one: Smith Apparent.

- Gooch Color2Gray was implemented in order to demonstrate that, it does not improve the quality of the results in practice, because of its inherent problems with the input parameter selection and its inconsistent spatial locality. Although its gradient-preserving nature could improve features discriminability.

- Grundland Decolorize was implemented in order to show the differences with our Multi-Image Decolorize, which as already mentioned is an adaptation of it.

In the rest of this section we give a detailed description of tested conversion algorithms. Then, we describe Multi-Image Decolorize (MID), after a description of the requirements analysis behind its design and development.

3.3.1 CIE Y

Assuming that the image is defined in the sRGB color space and has been linearized, the grayscale value Y_{xy} of the pixel in coordinates (x, y) is equivalent to the following weighted sum over the color values:

$$Y_{xy} = 0.212671R_{xy} + 0.71516G_{xy} + 0.072169B_{xy}\ . \tag{3.1}$$

3.3.2 Lightness Nayatani VAC

Assuming that the image is in the linearized sRGB space, the image is converted in the CIE L*u*v* space and the lightness thus calculated is altered in order to take into account the Helmoltz-Kohlrausch effect. The Lightness Nayatani VAC formula is:

$$Y_{xy} = L_{xy} + \left[-0.1340\,\mathrm{q}\left(\theta_{xy}\right) + 0.0872K_{Br_{xy}}\right] s_{uv_{xy}} L_{xy}\ , \tag{3.2}$$

where $s_{uv_{xy}}$ is a function of u and v which gives the chromatic saturation related to the strength of the H-K effect according to colorfulness, the quadrant metric $\mathrm{q}\left(\theta_{xy}\right)$ predicts the change in the H-K effect for varying hues and $K_{Br_{xy}}$ expresses the dependence of the H-K effect on the human eye's ability to adapt to luminance.

3.3.3 Gooch Color2Gray

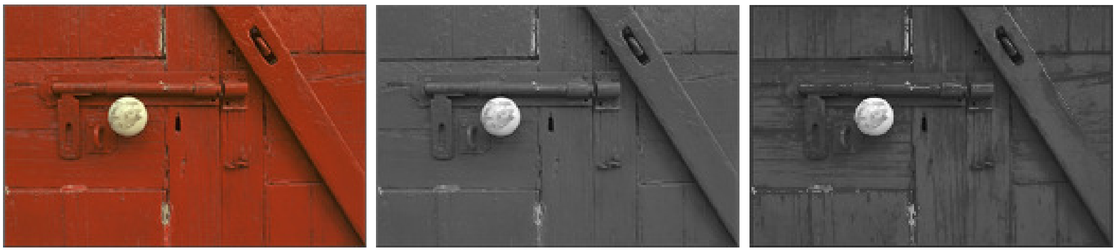
There are three steeps in the Gooch Color2Gray algorithm:

1. The color image is converted into a perceptually uniform CIE L*a*b* representation.

2. Target differences are computed in order to combine luminance and chrominance differences.

3. A least squares optimization is used to selectively modulate the differences in source luminance in order to reflect changes in the source image's chrominance.

The color differences between pixels in the color image are expressed as a set of signed scalar values δ_{ij} for each pixel i and neighbor pixel j. These δ_{ij} are signed distances based upon luminance and chrominance differences. The optimization process consists in finding grayscale values g such that all the differences $(g_i - g_j)$ between pixel i and a neighboring pixel j closely match the corresponding δ_{ij} values. Specifying δ_{ij} requires user interaction in order to obtain acceptable results. The



(a) Original colors (b) CIE Y (c) Gooch Color2Gray

Figure 3.3: An example of a Gooch Color2Gray conversion with a CIE Y reference on a 192×128 image and full neighborhood. The conversion took 106.5 seconds for Color2Gray, and 0.002 seconds for CIE Y.

output image g is found by an iterative optimization process that minimizes the following objective function, $f(g)$, where K is a set of ordered pixel pairs (i, j):

$$f(g) = \sum_{(i,j)\in K} \left((g_i - g_j) - \delta_{ij}\right)^2 , \tag{3.3}$$

g is initialized to be the luminance channel of the source image, and then descends to a minimum using conjugate gradient iterations [295]. In order to choose a single solution from the infinite set of optimal g, the solution is shifted until it minimizes the sum of squared differences from the source luminance values.

The user parameters, which need careful tuning, control whether chromatic differences are mapped to increases or decreases in luminance values, how much the chromatic can vary according to changes in the source luminance value, and how large is the neighborhood that is used to estimate the chrominance and luminance gradients.

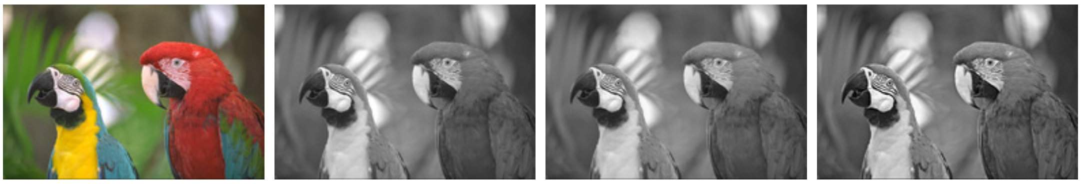
The computational complexity of this method is really high: $O(N^4)$, this can be improved by limiting the number of differences considered; e.g. by color quantization. A recent extension to a multi resolution framework by Mantiuk et al. [212] improves the algorithm's performance. In their approach, the close neighborhood of a pixel is considered on fine levels of a pyramid, whereas the far neighborhood is covered on coarser levels. This enables larger images to be converted.

Figure 3.3 shows a comparison between Color2Gray and CIE Y on a small image. Note that Gooch's approach (c) overemphasizes the small details of the wood texture with respect to both the original image (a) and the CIE Y (b).

3.3.4 Smith Apparent

The Smith Apparent algorithm can be summarized by the following two steps:

1. The color image is converted into grayscale using the Lightness Nayatani VAC technique explained in Section 3.3.2.



(a) Original colors (b) CIE Y (c) Nayatani VAC (d) Smith Apparent

Figure 3.4: An example of a Smith Apparent conversion, compared to CIE Y and to the algorithm's intermediate step Lightness Nayatani VAC.

2. The image contrast is enhanced using an unsharp masking which is adaptively weighted according to the chrominance information.

In the second step, to counter the reduction in local contrast in the grayscale image, unsharp masking is used to better represent the local contrast of the original color image. At this point, our implementation differs slightly from the technique described in Smith's paper [306]. While they use a general adaptively-weighted multi-scale unsharp masking technique [241], we simplify it by using a single-scale unsharp masking. This technique is adapted according to the ratio between the color and the grayscale contrast, so that increases occur at underrepresented color edges without unnecessarily enhancing edges that already represent the original.

For an example of the conversion, Figure 3.4 shows a comparison between Smith Apparent, Lightness Nayatani VAC and CIE Y on a colorful image. The figure also shows how Nayatani VAC (c) improves over CIE Y (b) in the hue change of the red parrot's wing and how Smith Apparent (d) restores the details of the image almost to its original quality (a).

3.3.5 Grundland Decolorize

The Grundland Decolorize algorithm has five steps:

1. The color image is converted into a color opponent color space.
2. The color differences are measured using a Gaussian sampling.
3. The chrominance projection axis is found by predominant component analysis
4. The luminance and chrominance information are merged.
5. The dynamic range is adjusted using the saturation information.

The first step takes a linear RGB image (with values in the [0 .. 1] range) and converts it into their YPQ representation. The YPQ color space consists in a luminance channel Y and two color opponent channels: the yellow-blue P and the red-green Q channels. The luminance channel Y is obtained with the NTSC Rec.601 formula, that is $Y_{xy} = 0.299R_{xy} + 0.587G_{xy} + 0.114B_{xy}$, while P and Q with $P = \frac{R+G}{2} - B$ and $Q = R - G$. The perpendicular chromatic axes support an easy calculation of hue $H = \frac{1}{\pi}\tan^{-1}\left(\frac{Q}{P}\right)$ and saturation $S = \sqrt{P^2 + Q^2}$.

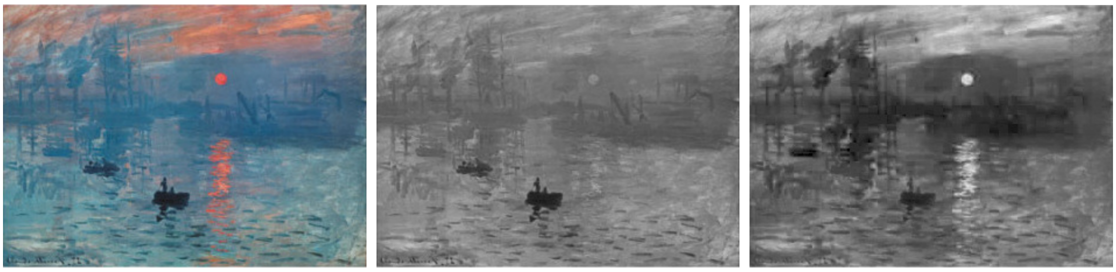
In the second step, to analyze the distribution of color contrasts between image features, the color differences between pixels are considered. More specifically, the algorithm uses a randomized scheme: sampling by Gaussian pairing. Each image pixel is paired with a pixel chosen randomly according to a displacement vector from an isotropic bivariate Gaussian distribution. The horizontal and vertical components of the displacement are each drawn from a univariate Gaussian distribution with 0 mean and $\frac{2}{\pi}\sigma$ variance.

To find the color axis that represents the chromatic contrasts lost when the luminance channel supplies the color to grayscale mapping, *predominant component analysis* is used. In the PQ chrominance plane, the predominant axis of chromatic contrast is determined through a weighted sum of the oriented chromatic contrasts of the paired pixels. The weights are determined by the *contrast loss ratio*[2] and the ordering of the luminance. Unlike the principal component analysis which optimizes the *variability* of observations, the predominant component analysis optimizes the *differences* between observations. The predominant chromatic axis aims to capture the color contrast information that is lost in the luminance channel. The direction of the predominant chromatic axis maximizes the covariance between chromatic contrasts and the weighted polarity of the luminance contrasts.

At this point (fourth step), the information on luminance and chrominance is combined. The predominant chromatic data values are obtained by projecting the chromatic data onto the predominant chromatic axis. To appropriately scale the dynamic range of the predominant chromatic channel the algorithm ignores the extreme values due to the level η of image noise. To detect outliers, a linear time selection algorithm is used to calculate the outlying quantiles of the image data. The predominant chromatic channel is combined with the luminance channel to produce the desired degree λ of contrast enhancement. At this intermediate stage of processing, the enhanced luminance is an image-dependent linear combination of the original color, which maps linear color gradients to linear luminance gradients.

The final step uses saturation to adjust the dynamic range of the enhanced luminance in order to exclude the effects of image noise and to expand its original dynamic range according to the desired degree λ of contrast enhancement. This is obtained by linearly rescaling the enhanced luminance to fit the corrected dynamic range. Then, saturation is used to derive the bounds on the permitted distortion. To ensure that achromatic pixels retain their luminance after conversion, the discrepancy between luminance and gray levels needs to be suitably bounded. The output gray levels are obtained by clipping the adjusted luminance to conform to the saturation dependent bounds.

[2] The relative loss of contrast incurred when luminance differences are used to represent the RGB color differences.



(a) Original colors (b) CIE Y (c) Grundland Decolorize

Figure 3.5: An example of a Grundland Decolorize conversion with a CIE Y reference.

The resulting transformation to gray levels is thus a continuous, piecewise linear mapping of color and saturation values.

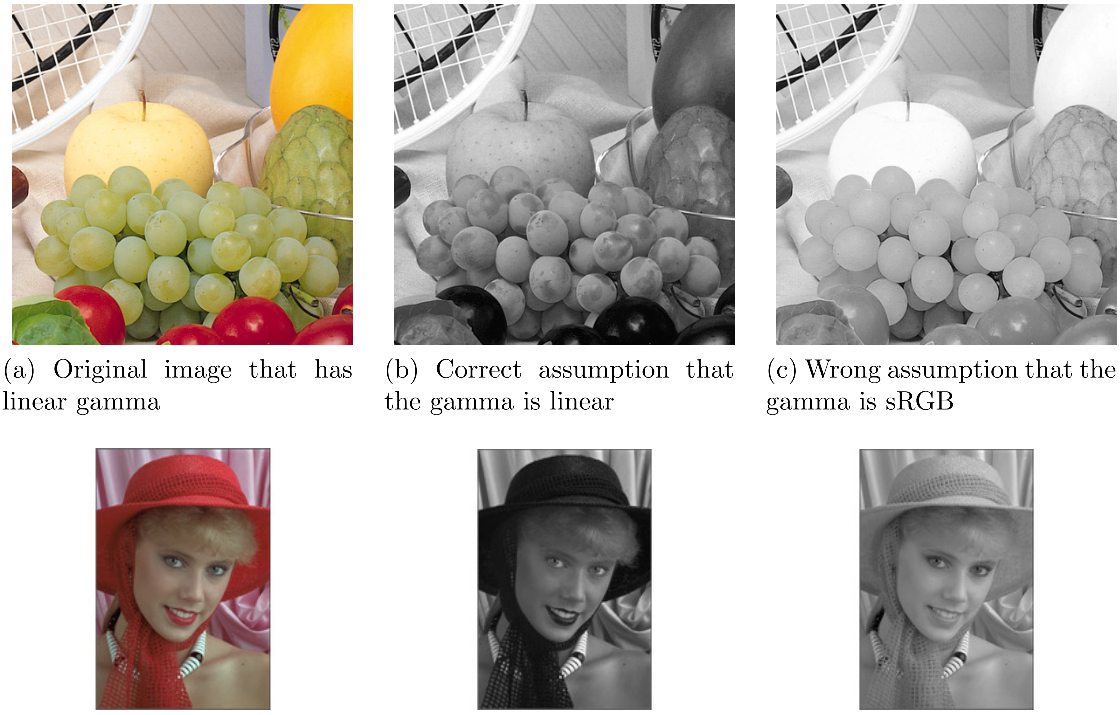
A comparison between Grundland Decolorize and CIE Y is shown in Figure 3.5. This image is “difficult” to convert into grayscale because most of the salient features are quasi-isoluminant with respect to their surroundings. The figure shows how Grundland’s approach (c) restores almost every feature of the color image (a) compared to a standard method such as CIE Y (b).

As already mentioned in Section 3.2.1, Grundland Decolorize is very sensitive to the issue of gamma compression. Figure 3.6 shows two examples of how an incorrect gamma assumption can decrease the quality of the results. A color image (a) has been linearized and then converted correctly assuming linearity (b) and wrongly assuming sRGB gamma compression (c). To show the complementary case, an sRGB image (d) has been converted wrongly assuming linearity (e) and correctly assuming its gamma compression (f). The loss of information is evident in the conversion which makes the wrong assumption: light areas (c) or dark areas (e) lose most of the features because the saturation balancing interacts incorrectly with the outlier detection. Moreover, the predominant chromatic axis is perturbed and consequently the chromatic projection no longer retains its original meaning. Note for example how the red hat and the pink skin (d), which should be mapped to similar gray intensities (f), are instead mapped to very different intensities (e).

3.4 Multi-Image Decolorize

In this section, we propose a theoretically-motivated grayscale conversion that is *ad-hoc* for the stereo and multi view stereo matching problem. Our conversion is a generalization of the Grundland Decolorize algorithm which simultaneously takes in input the whole set of images that need to be matched in order to be consistent with each other. In addition, two variants of the conversion are also proposed:

1. The first variant performs the original version of the proposed algorithm and



(a) Original image that has linear gamma

(b) Correct assumption that the gamma is linear

(c) Wrong assumption that the gamma is sRGB

(d) Original image that has sRGB gamma

(e) Wrong assumption that the gamma is linear

(f) Correct assumption that the gamma is sRGB

Figure 3.6: Two examples of right and wrong gamma assumptions with Grundland Decolorize.

then applies an unsharp masking filter in every image for enhancing feature discriminability.

2. The second variant is similar to the first but uses a chromatic weighted unsharp masking filter instead of the classic one.

3.4.1 Requirements analysis

Our goal was to design a conversion that transforms the image set by *preserving the consistency* between the images that are to be matched, i.e. the same colors in different images need to be mapped in the same gray values. In the meantime, it *optimizes* the transformation by exploiting the color information. To make our analysis clearer, we define the following *matching requirements*:

- *Feature Discriminability*: the method should preserve the image features discriminability to be matched as much as possible, even at the cost of decreased *perceptual* accuracy of the image[3].
- *Chrominance Awareness*: the method should distinguish between isoluminant colors.
- *Global Mapping*: while the algorithm can use spatial information to determine the mapping, the same color should be mapped to the same grayscale value for every pixel in the image.
- *Color Consistency*: besides Global Mapping, the same color should also be mapped to the same grayscale value in every image of the set to be matched.
- *Grayscale Preservation*: if a pixel in the color image is already achromatic, it should maintain the same gray level in the grayscale image.
- *Low Complexity*: if we consider the application of this algorithm in the context of multi view stereo matching, where a lot of images need to be processed, the computational complexity gains importance.

In addition, the proposed algorithm should be unsupervised, i.e., no user tuning is needed to work in a proper way.

3.4.2 Analysis of the state of the art

The Multi-Image Decolorize algorithm derives from a comprehensive analysis of the requirements described above. Our aim was to find the most suitable approach as a starting point for the development of our new technique.

[3] We will talk about the interesting correlations between perceptual and matching results in Section 3.5.7.

Bala Spatial was considered inadequate, because the spatial frequency based weighting of the importance of the H-K effect compared to the base lightness violates the Color Consistency and the Global Mapping requisites. As mentioned before, it was also susceptible to issues in chroma and lightness misalignment.

Gooch Color2Gray violates, above all, the low computational complexity requirement: its $O(N^4)$ computational complexity is too high for our application, and even Mantiuk's $O(N^2)$ improvement does not provide enough confidence in terms of quality versus complexity. Moreover, there are issues with the algorithm's dependence on parameters that could arbitrarily affect the grayscale mapping. This is good for artistic purposes, but is not useful with for our goals. Lastly, the gradient-based minimization process violates the Color Consistency, Global Mapping and Grayscale Preservation requirements.

Queiroz Invertible was unsuitable for our needs because it is designed for "hiding" the color information in "invisible" parts of the grayscale image, which does not improve feature discriminability in any way in terms of the standard conversions.

Rasche Monochromats has problems regarding the tradeoff between complexity and quality of the results because it quantizes colors. Moreover, it applies an energy minimization process which violates Color Consistency, Global Mapping and Grayscale Preservation requirements.

Neumann Adaptive is not appropriate for matching because image details and salient features may be lost by unpredictable behavior in inconsistent regions of the gradient field. Another issue is that this approach is aimed too towards human perceptual accuracy.

Grundland Decolorize complies to every requirement except the Color Consistency. Thus, we used this method as a starting point to develop our algorithm, extending it in order to respect such missing requirement.

The main issue with Alsam Sharpening and Smith Apparent is that, like Bala's approach, they violate our Color Consistency and Global Mapping requisites because of their unsharp masking like filtering of the images. This is a issue for our theoretical requirements. Hence, in this way, colors are mapped inconsistently between different parts of the images depending on the surrounding neighborhoods. Despite this, in some preliminary experiments with our implementation of the Smith Apparent conversion with respect to the Lightness Nayatani VAC, we found that the advantages of unsharp masking did improve the matching results. This is not surprising, because it is well known that the unsharp masking filter enhances the fine details of the image. Therefore, we also develop two variants of the Multi-Image Decolorize by adding an unsharp masking filter to the converted image.

We want to underline that the aforementioned requirements were sound in terms of improving of the matching task but, obviously, other ones can be defined to obtain improvements in the dense matching process.

3.4.3 The algorithm

Multi-Image Decolorize is an adaptation of the Grundland Decolorize algorithm which evaluates the whole set of images in order to match them simultaneously. To achieve this, we modified our implementation of Grundland's algorithm in order to execute each of the five steps simultaneously for each image in the set. Initially, this seems equivalent to the following procedure:

1. Stitch together, side by side, all images in the set in order to make one single big image.
2. Compute the Grundland Decolorize algorithm on the "stitched" image.
3. Cut back the grayscale version of the original images.

Nevertheless, this simple implementation would not work correctly because, in the Gaussian sampling step, near the common borders of the images a pixel could be paired with a pixel near the border of another image and the color differences estimation would be altered.

Instead, in order to achieve the desired result, the implementation performs each step of Grundland's algorithm on each image in the set before performing the next step, using the same accumulation variables for the predominant chromatic axis and for the quantiles of noise and saturation outliers. In this way, the matching requirements are fully applied to the set of images. In addition, the results benefit from the following transformation proprieties:

- *Contrast Magnitude*: the magnitude of grayscale contrasts visibly reflects the magnitude of color contrasts.
- *Contrast Polarity*: the positive or negative polarity[4] of gray level change in the grayscale contrasts visibly corresponds to the polarity of luminance change in color contrasts.
- *Dynamic Range*: the dynamic range of gray levels in the grayscale image visibly matches the dynamic range of luminance values in the color image.
- *Continuous mapping*: the transformation from color to grayscale is a continuous function. This reduces image artifacts, such as false contours in homogeneous image regions.
- *Luminance ordering*: when a sequence of pixels of increasing luminance in the color image shares the same hue and saturation, it will have increasing gray levels in the grayscale image. This reduces image artifacts, such as local reversals of edge gradients.

[4] That is the edge gradient.

- *Saturation ordering*: when a sequence of pixels with the same luminance and hue in the color image has a monotonic sequence of saturation values, its sequence of gray levels in the grayscale image will be a concatenation of at most two monotonic sequences.
- *Hue ordering*: when a sequence of pixels with the same luminance and saturation in the color image has a monotonic sequence of hue angles that lie on the same half of the color circle, its sequence of gray levels in the grayscale image will be a concatenation of at most two monotonic sequences.

In Fig. 3.7 we show how Multi-Image Decolorize is an improvement on Grundland Decolorize when applied on a image pair. While Grundland's approach gives better results when considering the images separately, its results are inappropriate when the pair of images is considered together. For example see the "L–G–1" corner of the cube:

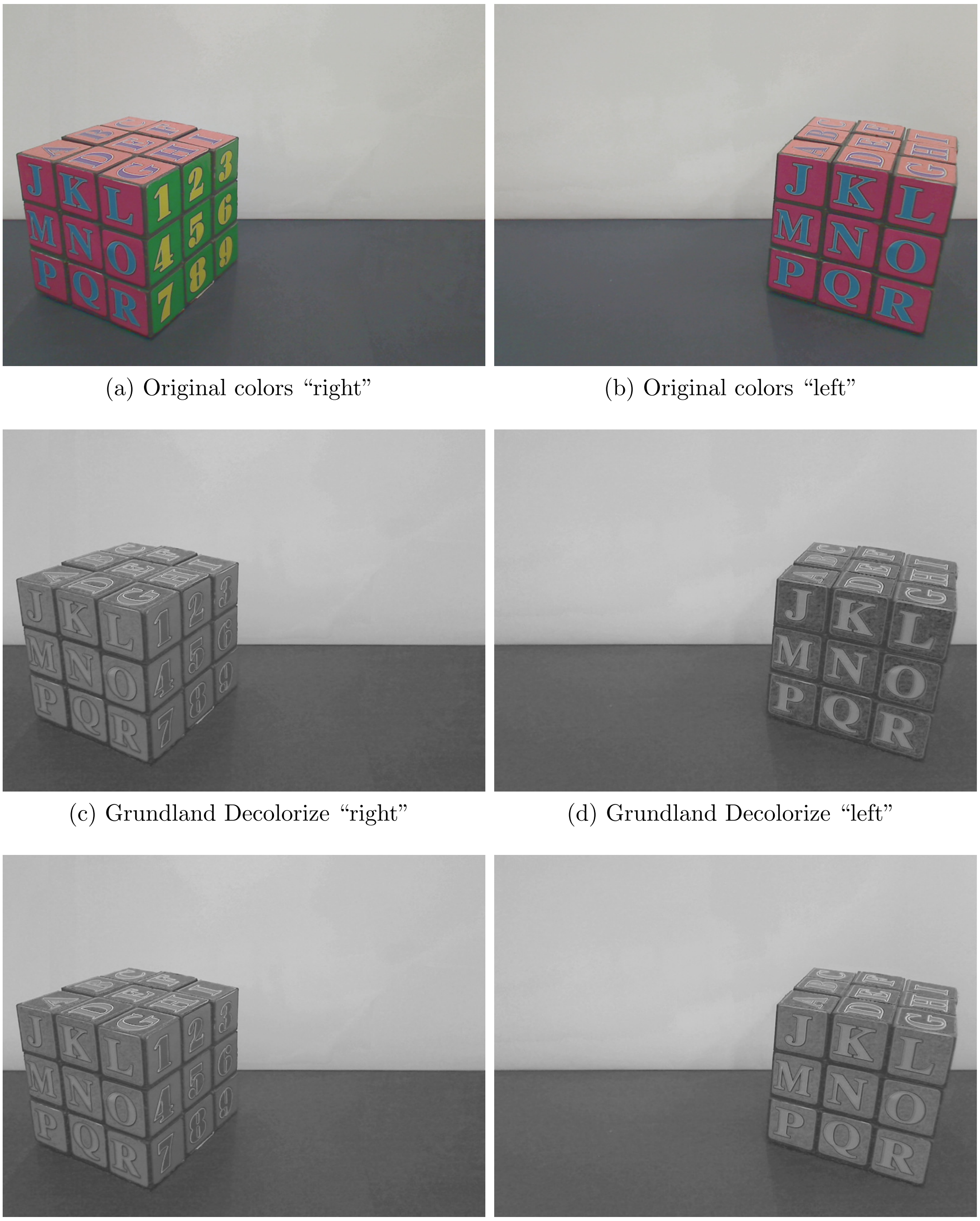
- In the "right" image (a), both Grundland (c) and the original version of Multi-Image Decolorize (e) have to cope with the presence of the green "1" side, and they obtain similar results.
- In the "left" image (b), where the green "1" side does not appear, Grundland (d) distinguishes the background of the "L" from the letter color better than the original version of Multi-Image Decolorize (f).
- If the "left" and "right" images were matched, the vast majority of the algorithms would have a greater probability of correctly matching the Multi-Image Decolorize pair (e) and (f) instead of the Grundland Decolorize pair (c) and (d).

This example emphasizes the differences of the two approaches and explains the advantages of our adaptation; whereas in real life scenarios these situations occur in a softer way, at least in stereo matching. In multi view stereo matching, where more images are involved, the benefits of a consistent mapping will be more relevant even in standard scenarios.

As Grundland Decolorize, Multi-Image Decolorize is also sensitive to alterations in the image gamma and, therefore, knowledge of the encoding of the starting image is essential.

3.4.4 First variant: classic unsharp masking

The technique described in the previous section converts input images consistently and appropriately. However, due to dimensionality reduction, the contrast may be reduced. To counter the reduction, we increased the local contrast in the grayscale image using the application of an unsharp masking filter on the converted image. Unsharp masking (*USM*) is the direct digital version of a well known darkroom analogic film processing technique [191] and is widely adopted in image processing [11] to improve the sharpness of a blurred image.



(a) Original colors “right”

(b) Original colors “left”

(c) Grundland Decolorize “right”

(d) Grundland Decolorize “left”

(e) Multi-Image Decolorize “right”

(f) Multi-Image Decolorize “left”

Figure 3.7: Difference between Multi-Image Decolorize and Grundland Decolorize in a stereo pair when chrominance changes significantly between the left and the right images.

3.4.5 Second variant: chromatic weighted unsharp masking

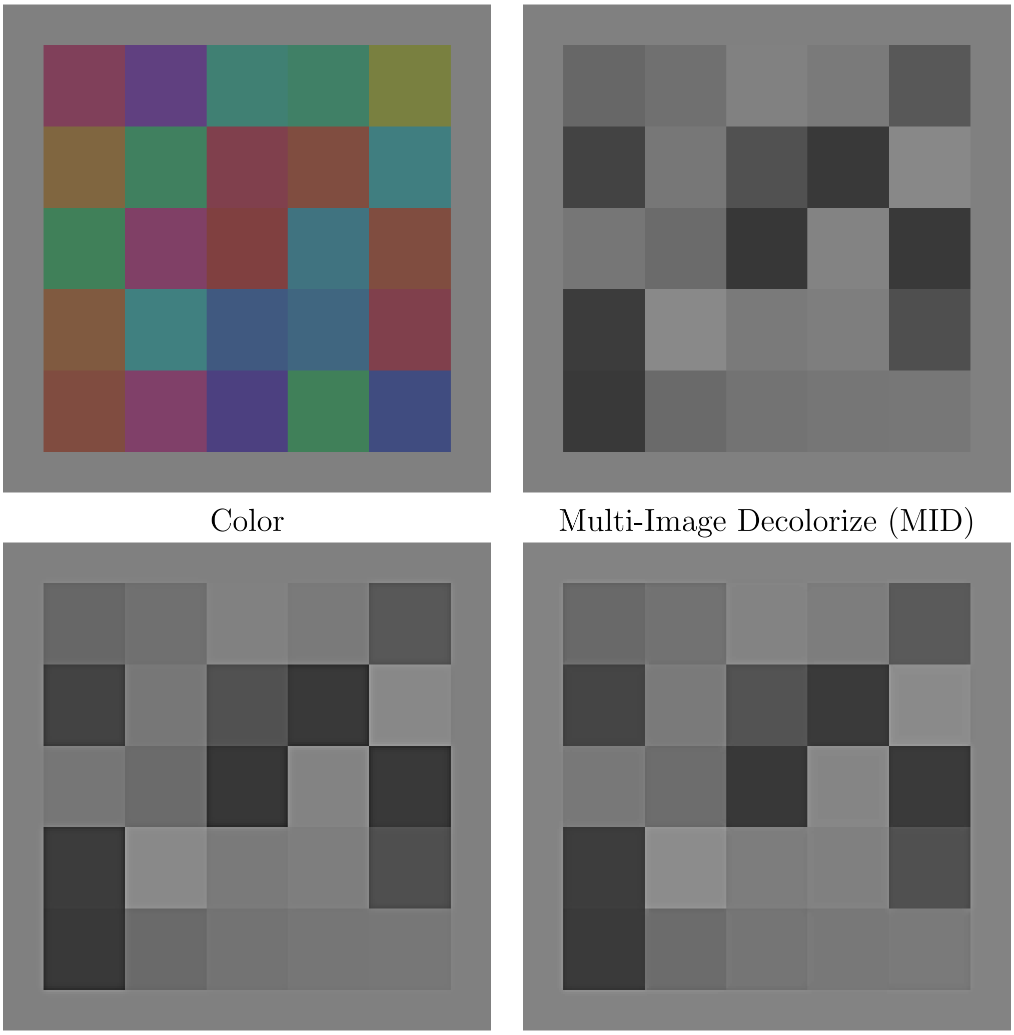
The idea of using USM filtering to improve the results derives also from the experimental performance of the Smith Apparent [306] technique, see Table 3.2. This is essentially a combination of the Lightness Nayatani VAC conversion with an ad-hoc USM filter. They adopted a chromatic-based adaptively-weighted version of the USM filter, which we simply call chromatic unsharp masking (*C-USM*), to counter the loss of *chromatic* contrast that derives from unaccounted hue differences. The technique is adapted according to the ratio between color and grayscale contrast, so that increases occur at underrepresented color edges without unnecessarily enhancing edges that already represent the original. Thus, this filter can represent the local contrast of original colors in a better way. We used a single scale simplification of C-USM, the same used in our implementation of the Smith Apparent method. The original implementation used in Smith's paper is multi-scale [306].

The effect of the local chromatic contrast adjustment is illustrated in Figure 3.8, where a nearly isoluminant color test pattern is converted into grayscale using the original version of the Multi-Image Decolorize, its first variant (MID with USM) and its second variant (MID with C-USM). The figure shows how the second variant gives different results compared to classical unsharp masking because it provides more contrast only where it is low in the conversion with MID and high in the color image; such as in the last squares of the bottom line. Where the contrast is good enough C-USM has a limited effect, for example, between the squares in the last two columns of the second and third rows.

3.5 Experimental Results

In this section, we will describe and discuss the results of the experimental evaluation of the grayscale conversions applied in the stereo matching context. We will show how the choice of the color to gray conversion preprocessing influences the precision of the reconstruction of a *Depth Map* (DM in the following) from a single stereo pair.

After the introduction of the StereoMatcher framework used to produce the results (Section 3.5.1), we will describe the various experimental components (Section 3.5.2). Since the number of results generated is too large to be discussed in full detail, we will first show a small subset in detail (Section 3.5.3). A comparison of Classic USM versus C-USM filtering (Section 3.5.4) is then presented and the general results are discussed (Section 3.5.6). Lastly we also compare the observed results with a recent study [53] of the *perceptual* performances of the various grayscale conversions (Section 3.5.7).



Color

Multi-Image Decolorize (MID)

First variant (MID with USM)

Second variant (MID with C-USM)

Figure 3.8: Different conversions of a nearly isoluminant color test pattern.

3.5.1 The StereoMatcher framework

Stereo matching is one of the most active research areas in computer vision. While a large number of algorithms for stereo correspondence estimation have been developed, relatively little work focused on characterizing their performance until 2002, when Scharstein and Szeliski presented a taxonomy, a software platform called StereoMatcher, and an evaluation [282] of dense two frame stereo methods. The proposed taxonomy was designed to assess the different components and design decisions made in individual stereo algorithms. The computation steps of the algorithms can be roughly aggregated as:

1. Matching cost computation

2. Cost (support) aggregation

3. Disparity computation / optimization

4. Disparity refinement

We used StereoMatcher to assess the impact of color to gray conversions. StereoMatcher is closely tied to the taxonomy just presented and includes window-based algorithms, diffusion algorithms, as well as global optimization methods using dynamic programming, simulated annealing, and graph cuts. While many published methods include special features and post processing steps to improve the results, StereoMatcher implements the basic versions of these algorithms (which are the most common) in order to specifically assess their respective merits.

Color processing in the StereoMatcher framework

The color is treated in the first step, which involves the computation of the matching cost. In StereoMatcher, the matching cost computation is the squared or absolute difference in color / intensity between corresponding pixels. To approximate the effect of a robust matching score [31, 281], the matching score is truncated to a maximal value. When color images are compared, the sum of the squared or the absolute intensity difference in each channel before applying the clipping can be used. If a fractional disparity evaluation is being performed, each scanline is first interpolated using either a linear or cubic interpolation filter [215]. It is also possible to apply Birchfield and Tomasi's sampling insensitive interval-based matching criterion [28], i.e., they take the minimum of the pixel matching score and the score at $\pm\frac{1}{2}$-step displacements, or 0 if there is a sign change in either interval. This criterion is applied separately to each color channel to simplify the implementation. In the words of the authors, this is not physically plausible; the sub-pixel shift must be consistent across channels. While this treatment has the advantage of using the color information, we believe it is inappropriate for our purposes, because when a color image is given it blindly sums the absolute or the squared differences. Moreover, when the sampling insensitive matching criterion is used, it may introduce significant inconsistencies.

Instead, we separated the color treatment from the matching cost computation by building a preprocessing tool to convert the original datasets and we used these resulting grayscale datasets as inputs for the StereoMatcher. As can be seen in the results, our approach sometimes provided an improvement compared to the results of the described color processing.

3.5.2 Description of the experiments

To thoroughly evaluate how the choice of different grayscale conversions affects the results computed by the StereoMatcher algorithms, we performed a large battery of tests. Thousands of error measures were computed, crossing different grayscale conversions with different StereoMatcher algorithms and with different datasets. Here, we only report the most representative and significant results. To describe the experiments we will catalog their components as follows:

1. *Datasets*: we used different datasets with ground truth, which are some of the standard datasets used in the Computer Vision community.

2. *StereoMatcher algorithmic combinations*: we used six different standard algorithms to obtain the depth maps.

3. *Classes of error measures*: we used two different kinds of measures of the computed depth maps errors.

4. *Areas of interest of the error measure*: we measured the errors in four different characterized parts of the depth maps.

5. *Grayscale conversions*: we used both the original color datasets and 11 different grayscale conversions.

This classification, detailed in the next sections, facilitates a comparison of the advantages and disadvantages of the grayscale conversions in terms of both the StereoMatcher algorithms and the peculiarities of the datasets.

The datasets

As just stated, the datasets employed in our experiments comes mainly from many subsequent works of StereoMatcher authors [152, 280], except one dataset, proposed by Nakamura in 1996 [229] and redistributed by them. These datasets are:

- The 1996 “tsukuba” dataset.

- Three 2001 datasets: “sawtooth”, “venus” and “map”

- Three 2005 datasets: “dolls”, “laundry” and “reindeer”.
- Three 2006 datasets: “aloe” and “cloth” “plastic”.

The datasets selected from these are: the “aloe”, “cloth”, “laundry”, “dolls” and “map”. The “map” dataset was originally in grayscale and was used only to validate the requirement that our conversion preserves the image quality when the colors were already achromatic.

We have no information on the gamma encoding of these datasets, however, using empirical measures of the image histogram distributions, we assume that only the datasets from 2006 are gamma compressed. Comparisons between the results of the linear-assuming and the sRGB-assuming versions of the Multi-Image Decolorize conversion seem to confirm this hypothesis.

The StereoMatcher algorithmic combinations

The dense stereo matching process takes two rectified images of a three dimensional scene and computes a disparity map, an image that represents the relative shift in scene features between the images. The magnitude of this shift is inversely proportional to the distance between the observer and the matched features. In the experiments, to obtain the computed depth maps, we used the following StereoMatcher algorithmic combinations:

- `WTA`: a *Winner Take All* disparity computation,
- `SA`: a *Simulated Annealing* disparity computation,
- `GC`: a *Graph Cuts* disparity computation.

The *Winner Take All* disparity computation algorithm simply picks the lowest matching cost as the selected disparity at each pixel. The *Simulated Annealing* and the *Graph Cuts* disparity computations are two iterative energy minimization algorithms that try to enhance the *smoothness term* of the computed disparity maps. We refer to [181] for the *Graph Cuts* algorithm and [282] for all the used algorithm and other StereoMatcher implementations. Each disparity computation was paired with either *Squared Differences (SD)* matching cost computation and *Absolute Differences (AD)* matching cost computation. As already explained in Section 3.5.1, the *AD* matching cost simply sums the absolute RGB differences between two pixels, while *SD* sums the squared RGB differences. Both cost computations truncate the sum to a maximal value in order to approximate the effect of a robust matching score. StereoMatcher does not allow computation of Normalized Cross Correlation (NCC) matching cost. For every algorithm, we used a fixed aggregation window, the spatial neighborhood considered in the matching of a pixel, and no sub-pixel refinements of the disparities.

The classes of error measures

To evaluate the performance of the various grayscale conversions, we needed a quantitative way to estimate the quality of the computed correspondences. A general approach to this is to compute error statistics with respect to the ground truth data. The current version of StereoMatcher computes the following two quality measures based on known ground truth data:

- `rms-error`: the root-mean-squared error, measured in disparity units.
- `bad-pixels`: the percentage of bad matching pixels.

The areas of interest of error measures

In addition to computing the statistics over the whole image, StereoMatcher also focuses on three different kinds of regions. These regions are computed by preprocessing the reference image and the ground truth disparity map to yield the following three binary segmentations:

- *textureless regions*: regions where the squared horizontal intensity gradient averaged over a square window of a given size is below a given threshold;
- *occluded regions*: regions that are occluded in the matching image, i.e., where the forward-mapped disparity lands at a location with a larger (nearer) disparity;
- *depth discontinuity regions*: pixels whose neighboring disparities differ by more than a predetermined gap, dilated by a window of predetermined width.

These regions were selected to support the analysis of matching results in typical problematic areas. We considered only the non-occluded (`nonocc`) regions since this kind of measure is the most significant one for our purposes. Hence, the other problematic areas, such as the textureless and occluded parts, could produce results that are not reliable in evaluating how the conversions could help the matching process.

The grayscale conversion

We executed the StereoMatcher algorithms and measured the various error measures for the following versions of the datasets:

1. Original color version, because we obviously needed a starting point to understand if the tested conversions would give worse, equal or even better results than the standard color approach.
2. CIE Y was chosen as the representative of “standard” grayscale conversions.

3. Sharp CIE Y, that is CIE Y followed by classic USM.
4. Chromatic Sharp CIE Y, that is CIE Y followed by C-USM.
5. Gooch Color2Gray, as the representative of the iterative energy minimization conversions.
6. Lightness Nayatani VAC as it is the starting point of Smith Apparent.
7. Sharp Lightness Nayatani VAC, that is Lightness Nayatani VAC followed by classic USM.
8. Smith Apparent, that is Lightness Nayatani VAC followed by C-USM, as the representative of the optimizing conversions that use spatial information.
9. Grundland Decolorize, as it is the starting point of our Multi-Image Decolorize technique.
10. The original version of Multi-Image Decolorize.
11. The first variant of Multi-Image Decolorize, that is Multi-Image Decolorize followed by USM.
12. The second variant of Multi-Image Decolorize, that is Multi-Image Decolorize followed by C-USM.

We computed these conversions for the five datasets just mentioned which gave a final number of $12 \times 5 = 60$ datasets. Thus, we ran StereoMatcher on 60 datasets using three algorithms (WTA, SA, GC) with two error measures (AD and SD) for a total of 360 tests. Due to the high number of tests done, in the next section, we detail a subset of the obtained results that are representative of the entire data collected. General consideration are presented in Section 3.5.6.

3.5.3 StereoMatcher results

First, the full details of three StereoMatcher algorithmic combinations with seven versions of the “laundry” dataset are shown. This dataset, whose original stereo pair can be seen in Figures 3.10(a) and 3.10(b) and whose true disparity map can be seen in Fig. 3.9, presents the typical situation in which our approach gives results that are similar or better than the usual color processing. The versions of the dataset that we show are:

- The original color version, in Fig. 3.10
- CIE Y, in Fig. 3.11,
- Gooch Color2Gray, in Fig. 3.12,

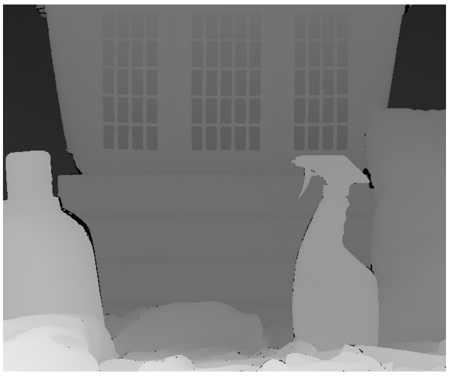


Figure 3.9: DM groundtruth for the “laundry” dataset

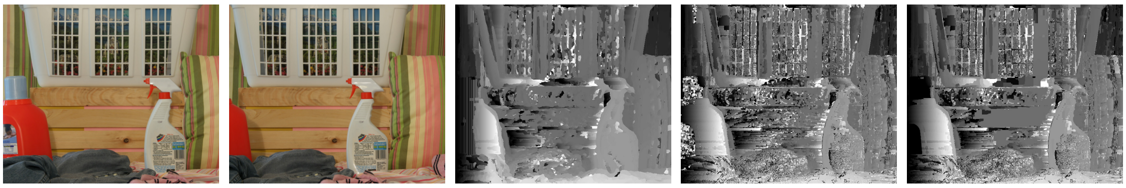
Table 3.1: Legend of histograms

Color	Version
▨	Original color version
🟥	CIE Y
🟫	Sharp CIE Y
🟫	Chromatic Sharp CIE Y
🟨	Gooch Color2Gray
🟩	Lightness Nayatani VAC
🟩	Sharp Lightness Nayatani VAC
🟩	Smith Apparent
🟦	Grundland Decolorize
🟦	Original Multi-Image Decolorize
🟦	First variant of Multi-Image Decolorize
🟦	Second variant of Multi-Image Decolorize

- Lightness Nayatani VAC, in Fig. 3.13,
- Smith Apparent, in Fig. 3.14,
- Grundland Decolorize, in Fig. 3.15,
- the original version of Multi-Image Decolorize, in Fig. 3.16.

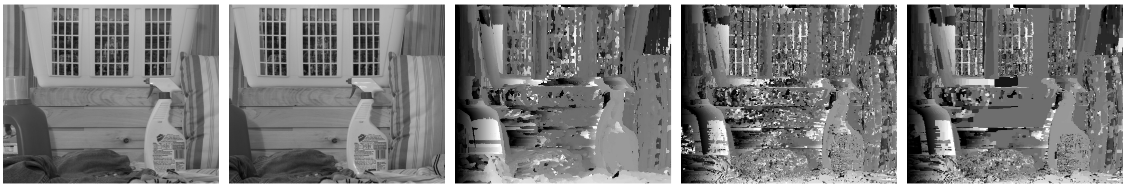
The error measures of the various versions of the datasets follow the color codes presented in the legend in Table 3.1. USM and C-USM variants are also included in the legend but are not shown here. However, we will use them in Section 3.5.4 for comparison purposes.

We only show the *Squared Differences*, because the results of the *Absolute Differences* and the *Squared Differences* variants of the algorithms used are really similar.



(a) Ref. Frame (b) Match Frame (c) DM from WTA (d) DM from SA (e) DM from GC

Figure 3.10: The “laundry” original dataset and three reconstructed DMs. DM groundtruth is in Figure 3.9.



(a) Ref. Frame (b) Match Frame (c) DM from WTA (d) DM from SA (e) DM from GC

Figure 3.11: The “laundry” dataset with CIE Y preprocessing and three reconstructed DMs.

More specifically, for every dataset version, we show:

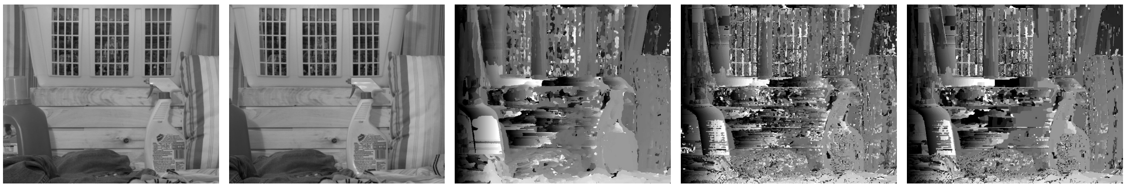
- the Reference Frame in subfigure (a),
- the Match Frame in subfigure (b),
- the disparity map for WTA in subfigure (c),
- the disparity map for SA in subfigure (d),
- the disparity map for GC in subfigure (e).

In Fig. 3.17 the histograms of the error measures are reported:

- Fig. 3.17(a) compares the errors when WTA is used,
- Fig. 3.17(b) compares the errors when SA is used,
- Fig. 3.17(c) compares the errors when GC is used.

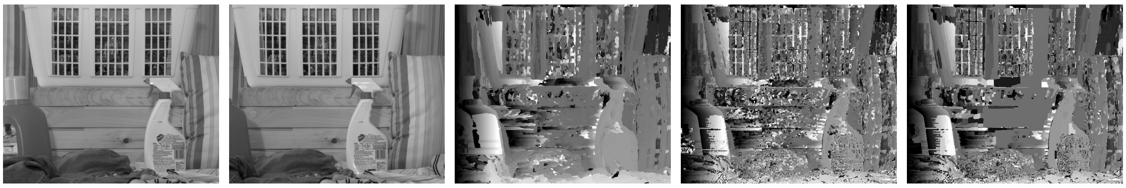
The same scale is used for each histogram.

This dataset contains elements, such as the background, which are really difficult for the employed algorithms. Our grayscale conversion is clearly the best one for this complex dataset, followed by Smith Apparent. When GC is used, Multi-Image Decolorize produces better results than color processing.



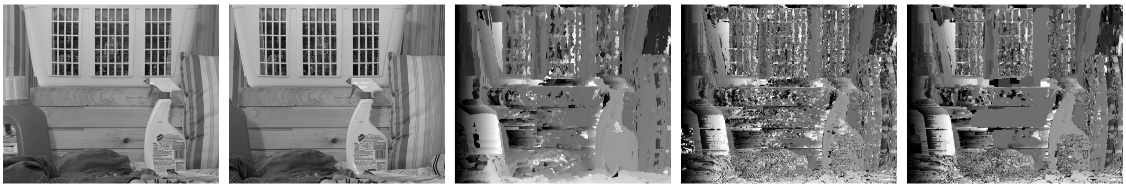
(a) Ref. Frame (b) Match Frame (c) DM from WTA (d) DM from SA (e) DM from GC

Figure 3.12: The “laundry” dataset with Gooch Color2Gray preprocessing and three reconstructed DMs.



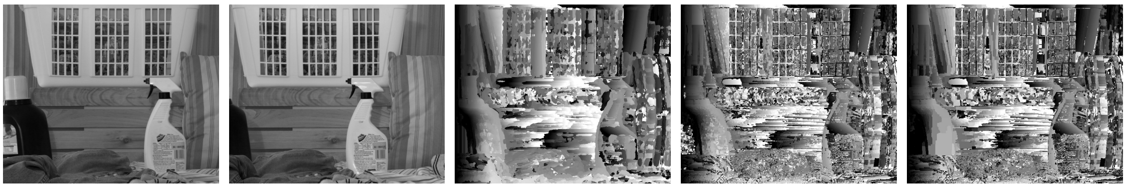
(a) Ref. Frame (b) Match Frame (c) DM from WTA (d) DM from SA (e) DM from GC

Figure 3.13: The “laundry” dataset with Lightness Nayatani VAC preprocessing and three reconstructed DMs.



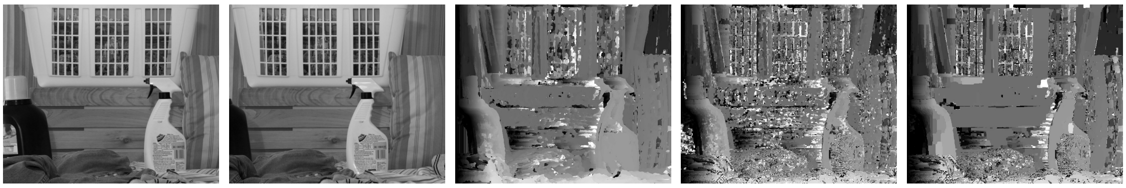
(a) Ref. Frame (b) Match Frame (c) DM from WTA (d) DM from SA (e) DM from GC

Figure 3.14: The “laundry” dataset with Smith Apparent preprocessing and three reconstructed DMs.



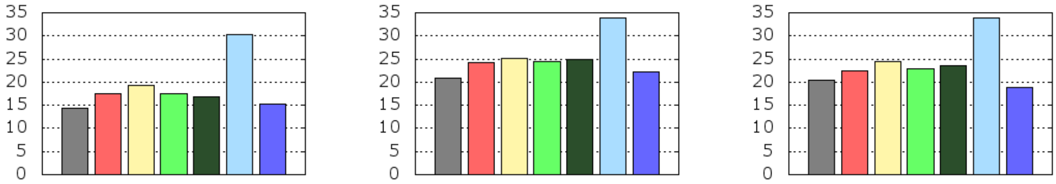
(a) Ref. Frame (b) Match Frame (c) DM from WTA (d) DM from SA (e) DM from GC

Figure 3.15: The “laundry” dataset with Grundland Decolorize preprocessing and three reconstructed DMs.



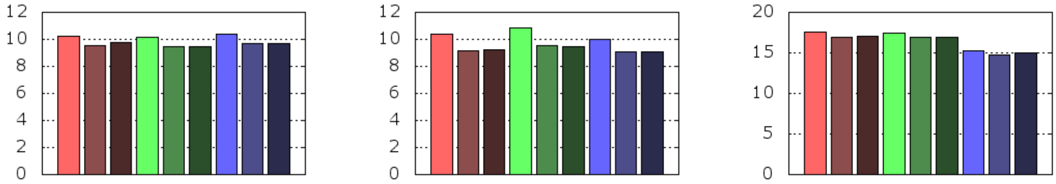
(a) Ref. Frame (b) Match Frame (c) DM from WTA (d) DM from SA (e) DM from GC

Figure 3.16: The “laundry” dataset with the original version of Multi-Image Decolorize preprocessing and three reconstructed DMs.



(a) WTA (b) SA (c) GC

Figure 3.17: `rms-error` of three StereoMatcher algorithms, in `nonocc` regions of the “laundry” dataset. The legend is in Table 3.1.



(a) The “aloe” dataset (b) The “cloth” dataset (c) The “laundry” dataset

Figure 3.18: `rms-error` of WTA, in `nonocc` regions of three datasets, which compares the non-unsharped versions of CIE Y, Lightness Nayatani VAC and Multi-Image Decolorize with the USM and C-USM versions. We recall that Smith Apparent is essentially a combination of the Lightness Nayatani VAC with a C-USM filter. The legend is in Table 3.1.

Table 3.2: `bad-pixels` of three StereoMatcher algorithms, in `nonocc` regions of four datasets, that compare the same versions of Fig. 3.17.

dataset	algorithm	Original color version ◼	CIE Y 🟥	Gooch Color2Gray 🟨	Lightness Nayatani VAC 🟩	Smith Apparent 🟩	Grundland Decolorize 🟦	Original Multi-Image Decolorize 🟦
aloe	GC	31.65%	21.99%	22.24%	22.51%	28.54%	26.84%	27.60%
aloe	SA	31.62%	27.17%	27.62%	28.16%	32.60%	31.37%	32.19%
aloe	WTA	10.06%	12.04%	12.21%	12.07%	9.90%	12.05%	11.64%
cloth	GC	36.32%	27.03%	32.62%	28.35%	29.80%	33.95%	32.46%
cloth	SA	36.11%	35.85%	38.77%	37.01%	36.05%	39.04%	37.80%
cloth	WTA	10.82%	16.02%	16.95%	16.68%	12.09%	16.60%	15.31%
dolls	GC	35.71%	33.14%	35.29%	34.04%	36.72%	38.72%	38.60%
dolls	SA	37.16%	40.45%	42.42%	41.13%	41.80%	45.68%	45.68%
dolls	WTA	20.02%	23.09%	23.96%	23.66%	20.80%	25.38%	25.17%
laundry	GC	61.65%	56.64%	59.98%	58.50%	60.03%	87.74%	48.13%
laundry	SA	67.53%	71.39%	70.54%	72.07%	72.64%	88.87%	66.48%
laundry	WTA	43.22%	50.67%	54.19%	51.15%	47.06%	80.86%	45.65%

Table 3.3: `bad-pixels` of WTA, in `nonocc` regions of four datasets, which compares the same versions in Fig. 3.18

method		aloe	cloth	dolls	laundry
CIE Y	🟥	12.04%	16.02%	22.88%	50.67%
Sharp CIE Y	🟫	9.86%	11.66%	20.30%	46.54%
Chromatic Sharp CIE Y	🟫	10.64%	12.08%	20.38%	46.72%
Lightness Nayatani VAC	🟩	12.07%	16.68%	23.03%	51.15%
Sharp Lightness Nayatani VAC	🟩	9.87%	12.21%	20.43%	47.37%
Smith Apparent (Chromatic Sharp VAC)	🟩	9.90%	12.09%	20.51%	47.06%
Original Multi-Image Decolorize	🟦	11.64%	15.31%	23.14%	45.65%
First variant of Multi-Image Decolorize	🟪	9.66%	11.47%	20.42%	42.62%
Second variant of Multi-Image Decolorize	🟦	9.75%	11.61%	20.61%	42.86%

Another evident fact is the poor performance of Grundland Decolorize. This is because in the Match Frame a big portion of the red bottle that was visible on the left of the Reference Frame is no longer visible, heavily changing the global chrominance of the image. By analyzing the images separately, Grundland Decolorize finds a different chromatic predominant axis of projection between the frames and thus assigns different grayscale values to the wood in the background. This causes the matching process in that region to fail, as highlighted in Figures 3.15(c), 3.15(d) and 3.15(e).

Table 3.2 also includes the `bad-pixels` error measures for non-occluded areas of the other four datasets with the `WTA`, `SA` and `GC` reconstruction. The table clearly shows that in general the best grayscale conversions are CIE Y, Smith Apparent and Multi-Image Decolorize, and often the Original color version has a larger error than one or more grayscale versions. CIE Y often gives the best results when aggregative algorithms such as GC and SA are used. These measures confirm the poor performance of Grundland Decolorize.

3.5.4 Classic USM versus C-USM

Here, we show how the choice of using either a classic USM or a C-USM after a grayscale conversion affects the matching results. To achieve this, we compare the results obtained for the following grayscale conversions:

- CIE Y:
  - in its original version,
  - with classic USM postprocessing,
  - with C-USM postprocessing;
- Lightness Nayatani VAC:

  - in its original version,
  - with classic USM postprocessing,
  - with C-USM postprocessing (that corresponds to the Smith Apparent method);

- Multi-Image Decolorize:
  - in its original version,
  - with classic USM postprocessing,
  - with C-USM postprocessing;

on three different datasets, "aloe", "cloth" and "laundry". The USM and the C-USM implementations are the same for each grayscale conversion. The reconstruction is performed by `WTA` and again we show the `rms-error` of non-occluded areas. In Figure 3.18, the histograms of the error measures are reported; Figure 3.18(a) compares the errors for the "aloe" dataset, Figure 3.18(b) for the"cloth" dataset, and Figure 3.18(c) for the "laundry" dataset. Please note that to improve readability between conversions in this case, the scale is *not* the same in every histogram.

From these results, two aspects can be underlined:

- Irrespectively of the dataset, both the USM and the C-USM versions perform better than the respective original algorithm.
- USM and C-USM have very similar performances.

To further confirm these observations, we also include in Table 3.3 the `bad-pixels` error measures for non-occluded areas of four datasets with a `WTA` reconstruction. To summarize, it is generally useful to apply unsharp masking filtering to improve stereo matching performances due to its enhancement of the fine details.

3.5.5 Computational time

Concerning the computational time of Multi Image Decolorize, we can state that for a stereo pair image with high resolution, e.g. 12 Megapixels, the overall time is in the order of a few seconds. In case of an extension of this algorithm to the multi-view case, this technique becomes a little more problematic, because the conversion could require a lot of time. The computational complexity of Grudland Decolorize is linear in the number of pixels on average, and $O(n \log(n))$ in the worst case. Consequently, the computational complexity of Multi Image Decolorize is $O(kn \log(kn))$ in the worst case, where k is the number of images to be processed. This is not a problem in the current Dense Stereo Matching approach, where $k = 2$, and is not likely to become a significant problem until the number of images is in the order of hundreds.

3.5.6 Summary of the results

Here, we discuss some general observations regarding the grayscale conversions tested and their relative performances.

- Although CIE Y is not as good as the optimizing conversions, it does have a very good *ratio* between complexity and performance. This is probably due to the robustness of its non-optimizing weighting of color values.
- Gooch Color2Gray gives poor results in our context, and it is computationally expensive;
- Lightness Nayatani VAC gives average results;
- Smith Apparent gives good matching results, due to its C-USM filtering. Its performance is often equal or better than Multi-Image Decolorize;
- Grundland Decolorize gives poor results and it is always worse than Multi-Image Decolorize. This is because it cannot cope with the image chrominance changes between the left and the right images;
- Multi-Image Decolorize is often one of the best non unsharp-masked grayscale conversions, followed by CIE Y.
- The fact that Grundland Decolorize has weak a performance while its multi-image extension is one of the best ones validates the theoretical assumptions of Sec. 3.4.1.
- For CIE Y, Lightness Nayatani VAC and the original version of Multi-Image Decolorize, both the USM and the C-USM filtering give *consistent* results, in most cases they improve the performance;
- There are not enough differences between USM and the C-USM filtering in terms of matching results to justify the adoption of the more complex C-USM in this field of application.

Other general considerations:

- StereoMatcher's *standard* approach to color information generally works well with respect to the tested grayscale conversions. However, in some cases, it performs similarly or even worse than a "good" grayscale conversion;
- given the constant improvements when USM filtering is used, we recommend its use in order to improve matching results;
- an assumption of the correct gamma compression is significatively important for all the optimizing conversions and it is critical for Grundland Decolorize. This is because the combination of this effect with Decolorize's lack of consistency can lead to unpredictable results;

- we can argue that the benefits of our grayscale conversion will be more evident when higher chromatic differences between images in the set are present.

3.5.7 Matching and perception

There are some interesting similarities between our results and an external study of the perceptual performances of many grayscale conversions that we used in this work.

To our knowledge, the study presented in Čadík et al. [53] is the first perceptual evaluation of modern color to grayscale conversions. In this paper, they presented the results of two subjective experiments in which a total of 24 color images were converted to grayscale using seven grayscale conversion algorithms and evaluated by 119 human subjects using a paired comparison paradigm. The grayscale conversions perceptually compared were: CIE Y, Bala Spatial, Gooch Color2Gray, Rasche Monochromats, Grundland Decolorize, Neumann Adaptive and Smith Apparent. About 20000 human responses were used to evaluate the accuracy and preference of the color to gray conversions. The final conclusions of this work have some similarities with our study. In both studies:

- Grundland Decolorize and consequently our Multi-Image Decolorize adaptation is one of the best conversions.

- Smith Apparent is one of the best conversions.

- CIE Y performs well notwithstanding its simplicity.

Obviously, the role of perception in machine vision algorithms is out of the scope of this work. However, it is an interesting point that stereo matching results are somewhat correlated to human perceptual preferences.

Chapter 4

Multi-view processing for image-based appearance reconstruction

On the side of ***multi-view processing for image-based appearance reconstruction****, we present an image pre-processing algorithm, that basically is a shadow removal algorithm. This helps increasing the quality of color mappings from images to a 3D surface. The result of an outdoor 3D scanning acquisition campaign is usually an accurate 3D model of the site, but in most of the cases the quality of the color acquired by the scanner is not satisfying. Alternative solutions, as the projection on the object of a photographic dataset captured in a different stage, are still dependent on the quality of the acquired images. The short time for the acquisition campaigns and the weather conditions often force the shooting of images taken under a strong direct sun illumination. The presence of shadows generates colored models of poor quality. The use of georeferencing of the 3D model and of time information from the image data allows for a sun position estimation. This can be exploited in a color preprocessing approach for 2D/3D color mapping, through computation of virtual shadows, segmentation of shadowed regions from the input images, and assignment of "bad quality" to shadowed regions in images. This quality assessment can be used to prevent use of inciding pixels in subsequent texture synthesis when possible, but also for removal of the shadows from the input images in order to gracefully provide color data where the only color source for part of the surface comes from shadowed regions. Using this kind of approach, outdoor sites can be acquired producing a high quality color information together with an accurate geometric measurement.*

4.1 Introduction

The use of three-dimensional data in the context of Cultural Heritage is becoming more and more popular. While the acquisition hardware is still quite expensive, the commercial and freeware tools to process the acquisitions can provide ways to visualize and analyze complex data.

As explained in Sec. 2.1.4, an important aspect of a three-dimensional reconstruction is the surface appearance. We have seen that this aspect appears to be much more complex with respect to the geometry reconstruction: especially for large artifacts, the standard setups for the acquisition of material properties are usually too complex for practical applications. As previously stated, an alternative approach is to map color information from groups of images. Basically, if the light position is unknown, the lighting artifacts are projected on the 3D model as if they were color information. Therefore, controlled lighting is usually necessary to limit the presence of artifacts such as shadows or highlights.

When the object to be acquired is large and outdoor, lighting can be rarely controlled. A cloudy day usually provides an almost perfect environment (i.e. diffuse lighting and no shadows), but the scanning campaign has to be usually completed in a short time, and in some places (e.g. Africa, Asia and South America) cloudy days are quite rare.

Since the acquired images can present strong artifacts (i.e. hard shadows and highlights), they need to be detected and subsequently removed. In this chapter, we present an approach to improve the quality of color projection of images taken under direct sun illumination. The sun position at the moment of the photo shooting can be obtained if the three-dimensional data are geo-referenced, and the time and date of the photo is known.

If the sun position is known, then:

- the image alignment process can be speeded up;
- the shadows positions in the image can be automatically detected.

Then, it is possible either to try to remove the shadows and used the unshadowed image set directly in any color projection framework, or to adjust weights on the shadows parts in weighting blending scheme such as the one by Callieri et al.[54]; that is the color mapping framework we used here to obtain the colored model.

The final colored 3D model will present a more realistic appearance. Finally, it could be possible to apply the present approach to a previously acquired dataset.

4.1.1 Contributions

The main contribution is a method that uses the sun position to improve the color projection pipeline. If the sun position associated to an image can be recovered, all the steps of the pipeline (image alignment, image processing and color projection) can be made more robust and reliable.

Moreover, the entire process can be implemented in a completely automatic way, and integrated in existing frameworks.

The proposed approach can be usefully applied in the field of Cultural Heritage, because it does not add any additional effort during an acquisition campaign, where all the needed data like GPS coordinates and images are already acquired as a routine.

4.2 Modeling the sun: light direction estimation

The direct sunlight usually represents an issue for photographers, due to the strong lighting on exposed surfaces and the hard shadows produced.

But at the same time, due to its distance with respect to the earth, the sun light source can be easily approximated as a directional one, where the light direction is the same throughout the scene. Hence, to model the sun light, only the angle between it and surfaces are needed to be known.

During the acquisition of the images, the sun position and other data can be acquired by using ad-hoc devices [326, 72]. Recently, other approaches try to estimate the sky environment directly from images [188, 189].

If no acquisition device is available, and accuracy is needed, several simple online tools [43, 277, 37] can calculate the sun position, which is usually expressed with two angles:

- the *azimuth*, which is mostly defined as the angle along the horizon, with zero degrees corresponding to North, and increasing in a clockwise fashion;
- the *elevation*, which is the angle up from the horizon.

The inputs, needed for the calculation, are data about the site location (e.g. latitude and longitude) and the date and time when the image was taken. The site position can be acquired by storing the GPS coordinates of some reference points: with at least three points, the corresponding 3D model can be geo-referenced so that its orientation is aligned to the north direction. The date and time can be easily retrieved from the EXIF metadata of the image. In this way, all required inputs are available.

Alternatively, there is a manual procedure that can be used to estimate the sun light direction. The approach is similar to the one used by Dellepiane et al. [88] to estimate the flash light position: first, the user needs to align the image on the 3D model. Then, it is necessary to indicate on the image a point on the 3D model, like a corner or a strong geometric feature, and its corresponding projected shadow on the 3D model. If the image is aligned to the model, this identifies two points in the space which should define the sun direction for that image. Indicating several couples of points and averaging the resulting direction could lead to an accurate enough estimation.

Once that the sun direction is known, both the alignment and the projection phases can be enhanced in order to produce better results: the next Section will show how this is implemented.

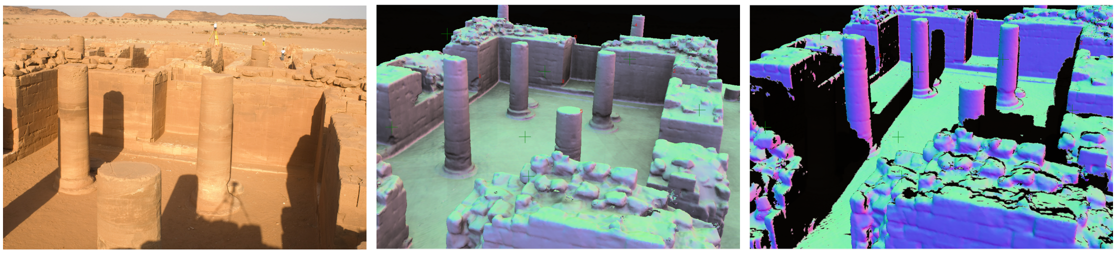


Figure 4.1: Left: an image with sun light illumination. Center: the rendering of the corresponding model using ambient occlusion and normal maps. Right: the rendering of the corresponding model with normal maps and shadows generated by the estimated sun light direction.

4.3 Shadows detection and removal

The sun direction estimation transforms a generic color projection dataset in a dataset with a controlled light setup. This greatly enhances the possibilities in most of the steps of the color projection pipeline: image alignment, image correction and color projection. The next subsections will show how this can be easily exploited.

4.3.1 Image alignment using sun direction information

When dealing with an un-calibrated set of images, the preliminary step of image alignment can be quite difficult and time consuming. While the semi-automatic solution [110] proved to be robust, it is time consuming if applied to a set of images. A more recent technique [74] uses mutual information to fit a illumination related rendering of the 3D model to the image. In the original idea, the most robust rendering was a combination of normal maps, related to directional illumination, and ambient occlusion, accounting for diffuse component. In our case, since the light direction is known, the ambient occlusion can be substituted with a shadow mapping on the 3D model. In this way, the shadows are in the same position as in the image, and the convergence of Mutual Information maximization is faster and more precise. Figure 4.1 shows an image and two renderings of the corresponding models using ambient occlusion and shadows generated with the estimated sun light direction.

The second type of rendering is clearly much more correlated to the appearance of the real object, so that the image alignment process is much more fast and robust.

4.3.2 Shadow masks creation and image correction

Once that the image alignment is complete, all the needed datum for the color projection are available. But before the final step, the light direction information can be further used. First of all, in a similar fashion to the shadow detection feature of [88], it is easy to create a shadow mask that indicates which portions of each image

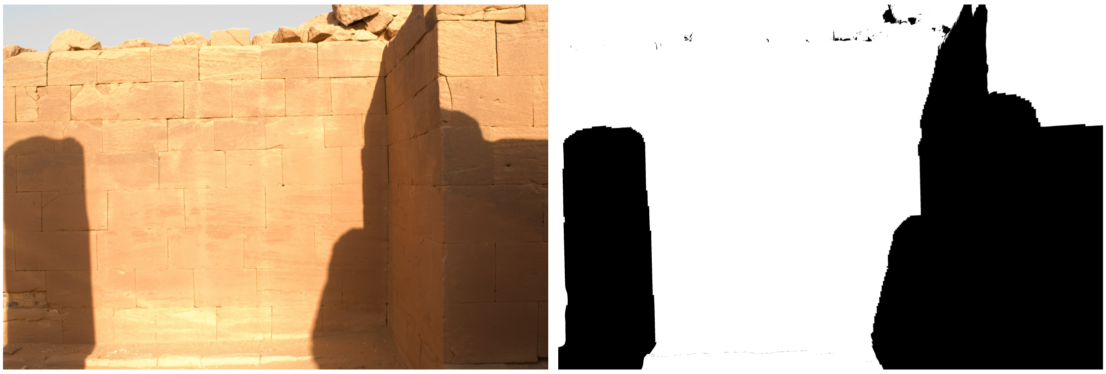


Figure 4.2: Left: an image with direct sun light illumination. Right: the shadow mask extracted after image alignment to the digital 3D model.

of the photographic dataset are in shadow. The comparison between a rendering from the image point of view and a rendering from the light direction shows that all points can be considered to be under shadow if they are visible from the camera location, but not visible from the light direction. Figure 4.2 shows an image and the corresponding shadow mask: the silhouette of the main shadows is extracted in a very accurate way.

The shadow masks can be used during the projection phase, in order to mask the contribution of the portions of the image to the final color. However, they can be also valuable to try to pre-process the images. In the last years, several techniques to remove shadows from an image have been proposed [7, 296, 219, 103, 102, 111], but the shadow detection is usually a semi-automatic process [296, 219], although some quite robust techniques have been proposed [103, 102, 111]. In our case, the possibility to calculate the shadow maps permits to skip the initial step, so that the image correction can be made in a completely automatic way.

Our approach is inspired by Fredembach and Finlayson [111]. The main difference with this method is the fact that we already have the shadow maps. The shadow removal procedure follows these steps:

- Coarse shadow edges locations are identified. A portion of the image around each edge location is considered, since the quality of shadow maps is not always perfect; see later for a discussion on this drawback.

- For every edge zone, the maximum intensity difference (offset) between the illuminated and the shadowed part is obtained. In order to remove noise, a lowpass filtered version of the images is used. Moreover, if the portions in shadow and in light of the zones are not uniform, the zone is not taken into account in order to prevent from inaccurate corrections.

- The obtained offsets are interpolated with a pull-push algorithm [136] to obtain smooth non-constant values. These offsets are summed to the shadow region.

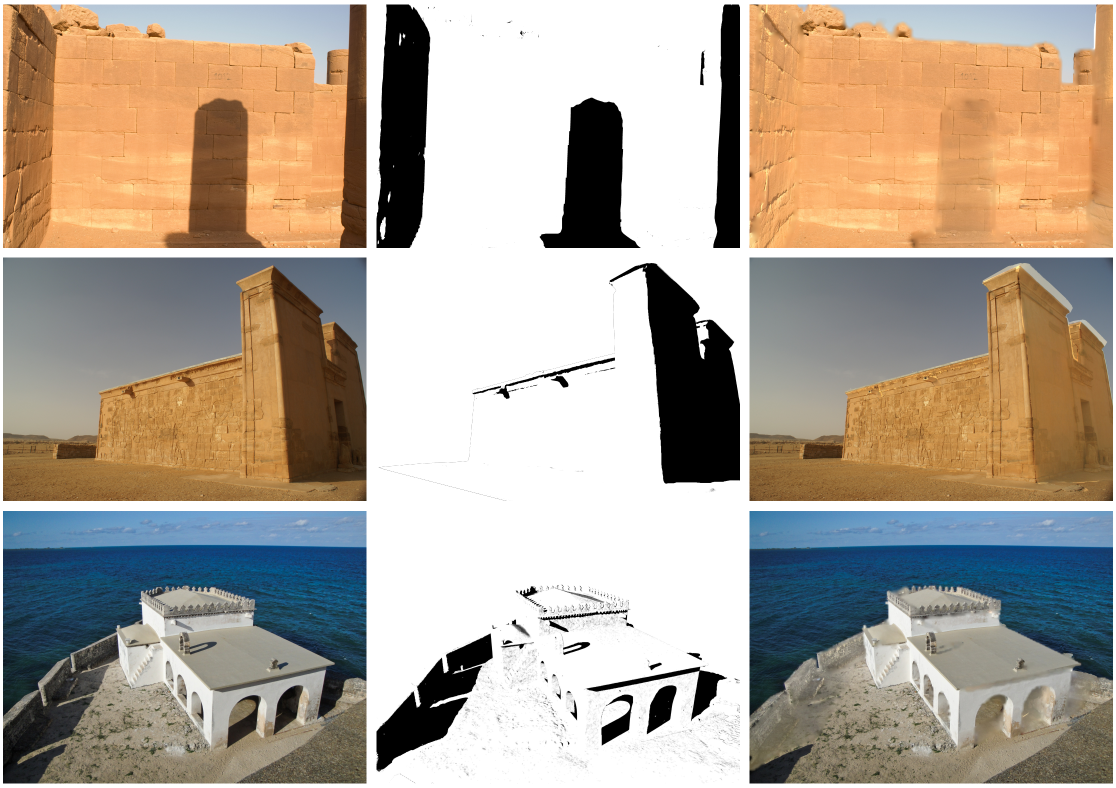


Figure 4.3: Left column: an image with sun light illumination. Center column: the shadow mask used for image correction. Right column: the result of shadow removal.

- Finally, the shadow edges are deleted and recovered by pull-push interpolation, to ensure minimization of the errors.

The algorithm is almost automatic except for a parameter, which defines the size of the edge zone where the shadow edges need to be found. This parameter is linked to the quality of the shadow mask, and it can be changed in order to deal with difficult cases, where for example the three-dimensional geometry is not accurate enough to obtain good shadow masks.

Figure 4.3 shows two examples of image processing: the first column shows the original images, the central column shows the shadow masks used for correction, the right column shows the results. As it can be noted, the original color of the object is reconstructed with sufficient accuracy, and also the details of the surfaces is preserved. Some artifacts are present only on the border of the shadows. The third examples shows a more complex case, with different colors throughout the scene. The shadow removal obtains acceptable results. A portion of shadow was not removed because it was not detected, because it was generated by some structure (a wall portion) which was not part of the digital 3D model.

These artifacts can be generated by the small errors of the shadow maps, which are sometimes generated by three-dimensional models not accurate enough to reproduce the shadows in the images. The correction of these artifacts would require some intervention by the user, in order to correct the shadow maps or weight the shadow removal in a different way. In the context of the proposed system, it was decided to preserve the fully automatic approach. This was achieved by taking into account that the portions of the images, which contain artifacts, can be known in advance. Hence, using the quality weighting factor in the color projection phase (see next Section), a very low quality value can be assigned to these zones. Therefore, they are used only when no contribution comes from other images.

4.4 Color projection

The results obtained in the previous sections can be exploited in the final step of color projection. Hence, the corrected images can be used in order to obtain a more coherent colored model. Moreover, the shadow masks extracted in the previous section can be integrated in weighting blending scheme, we recall that we used here the framework proposed by Callieri et al. [54].

For example, a lower weight can be assigned to the portions of the images which underwent the shadow removal, so that they could be used only when the contribution of other images is not good enough.

4.5 Results

The proposed approach was applied on a number of datasets, mainly coming from the African Heritage. Especially in these cases, the acquisition campaign are performed during the dry season, when the weather is usually very sunny. As a result, most of the photographic campaigns are performed under strong sun light, presenting hard shadows in almost all images. Starting from set of multiple images, the results obtained with the classic projective approach were compared with the proposed technique.

A first example is shown in Figure 4.4, where six images have been mapped in a portion of the ruins of a temple. Since all the photos were acquired in a short time, the position of the shadows did not change noticeably. While the rendering is obtained using a soft diffuse lighting, hard shadows independent from the light environment are noticeable on the scene. Even the blending approach of Callieri et al. [54] is not able to mask them, because some portions of the surface are covered only by images in shadow. The resulting three-dimensional model presents strong artifacts as can be seen in the left part of Figure 4.4.

An alternative solution could be to perform photographic campaigns in different times of the day, and then to mask the images by not projecting the parts in shadow. However, this operation would be time consuming, and the coherency of the final

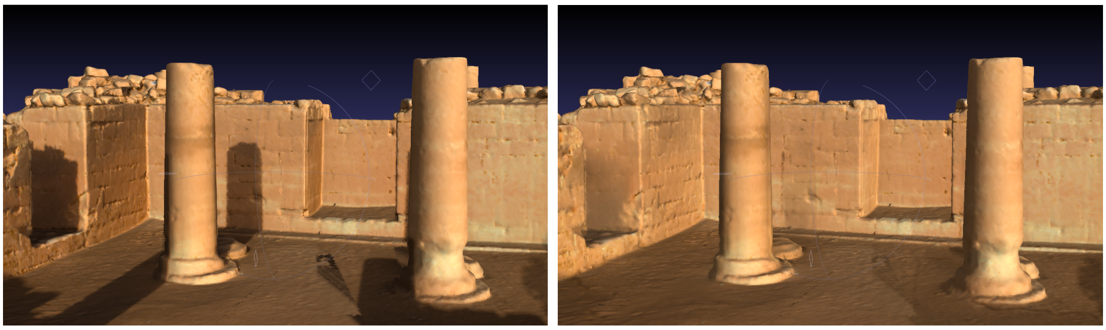


Figure 4.4: Left: a rendering of the 3D model without shadows removal. Right: a rendering of the 3D model with shadows removal.

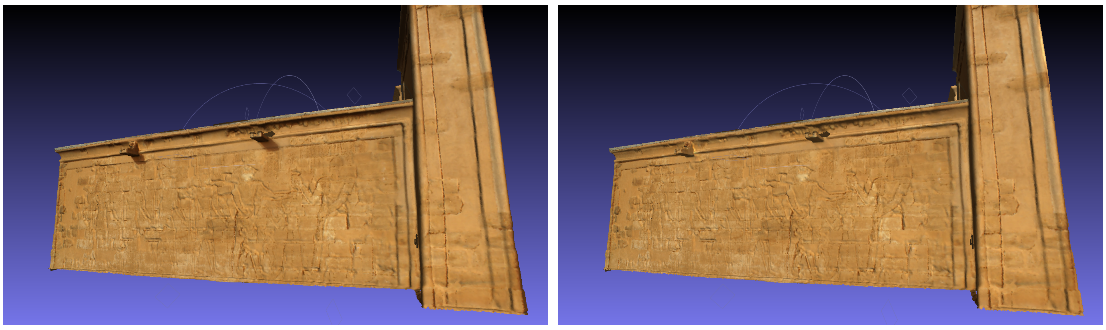


Figure 4.5: Left: a rendering of the 3D model without shadows removal. Right: a rendering of the 3D model with shadows removal.

color is not guaranteed. Nevertheless if the images are processed using the proposed approach, the resulting model shows a much more realistic color, as shown in the right part of Figure 4.4. Only a few artifacts are still remaining, but the model can be re-illuminated with a higher degree of realism.

Figure 4.5 shows a second example, where the original shadows are smaller but still noticeable. In this case, four images projected on a portion of a temple, the shadows are removed from the final three-dimensional model, so that the appearance and the navigation result to be more realistic. The method was applied on several other test cases, resulting in evident improvements on the meshes.

The main limitations of the approach are related to the accuracy of the initial dataset: the shadow masks could not be precise if the 3D model is not accurate or the sun direction is not appropriately estimated. Moreover, the accuracy of the geometry influences also the color projection phase. Another intrinsic limitation is that if the shadows on an image were generated by external objects (e.g. people and parts of the site which have not been acquired) these artifacts cannot be corrected by our approach. In this case, the user shall apply semi-automatic image-based approaches.

Chapter 5

Multi-view processing for 2D/3D registration

On the side of ***multi-view processing for 2D/3D registration****, we present a large scale image-to-geometry registration system, that can localize and calibrate an unknown image against a set of multi-view datasets with associated 3D models. The system is able to recognize the site that has been framed, and calibrate it on a preexisting 3D representation. Furthermore, this system is characterized by very high accuracy and it is able to validate, in a completely unsupervised manner, the result of the localization. Given an unlocalized image, the system selects a relevant set of pre-localized images, performs a Structure from Motion partial reconstruction of this set. Then, it obtains an accurate camera calibration of the image with respect to the model by minimizing distances between projections on the model surface of corresponding image features. At this point, the obtained calibration is compared with the one from the structure from motion (suitably translated in the model coordinate frame) using visual similarity metrics in order to validate the results. The reached accuracy is enough to seamlessly view the input image correctly super-imposed in the 3D scene. The algorithm has been demonstrated in a real scenario of digital support for tourism: a "virtual visit" of a place can be a valuable experience before, during and after the experience on-site; the completely automatic algorithm allows a tourist to virtually embed its own photographs in a digital reconstruction of the places who has been visited.*

5.1 Introduction

Automatic image localization is an active research field in computer vision and computer graphics, with many important applications. This has become especially important given the potentials of all images coming from the web community. Traditional localization solutions, e.g. Global Positioning System (GPS), may present issues in certain urban areas or indoor environments, or may not be accurate enough.

Moreover, both the position and the orientation of the camera could be a valuable source of data. Alternatives, such as inertial drift-free systems, are too expensive to be applied on a large scale. In this case, the only class of solutions realistically feasible today is the use of image-based localization systems. Many aspects of the automatic image localization problem have been independently tackled, and tremendous advances have been obtained in recent years.

Here, we are interested in the automatic user localization through the use of digital consumer cameras or smart phones to support information services for tourists. In particular, we cope with a specific image localization scenario, that, to our knowledge, has never been faced in literature: exploiting pre-existing high quality 3D models of the photographs' environment for performing an offline, fully automatic, precise and unsupervised image localization. We aim to obtain such an accuracy to allow a *seamless* view immersion into the 3D scene by projecting the photos on the 3D models. This scenario differs from apparently similar ones, like the SLAM problem [307], both for the significantly higher level of precision required and, most importantly, for the intrinsically heterogeneous nature of the input images.

High accuracy allows to re-visualize the picture of the tourist in PhotoCloud [45], a visualization system which shares some similarities with Photo Tourism [309]. This system was developed during the work of this thesis and it is described in-deep in the next Chapter. Note that PhotoCloud is one of the main goals of an ongoing project related to tourism and valorization of artistic sites.

5.1.1 Contributions

The proposed system effectively merges solutions from image retrieval, Structure from Motion and 2D/3D registration. In this context, our contribution is twofold:

- an *image-based localization algorithm*, capable to obtain very high accuracy by exploiting pre-existing high quality 3D models of the locations of interest,

- an *unsupervised validation algorithm*, that guarantees to present only correct results to the user.

The developed system works by exploiting a dataset of pre-aligned digital photographs on 3D models of the locations of interest.

Our work shares some similarities with the methods of Irschara [168] and Sattler [278], described with other related works in Section 2.2. The novelty stands in the use of a more advanced image retrieval algorithm, the exploitation of 3D geometric information, that is not dependent on the photographic dataset, and the validation through an unsupervised validation algorithm.

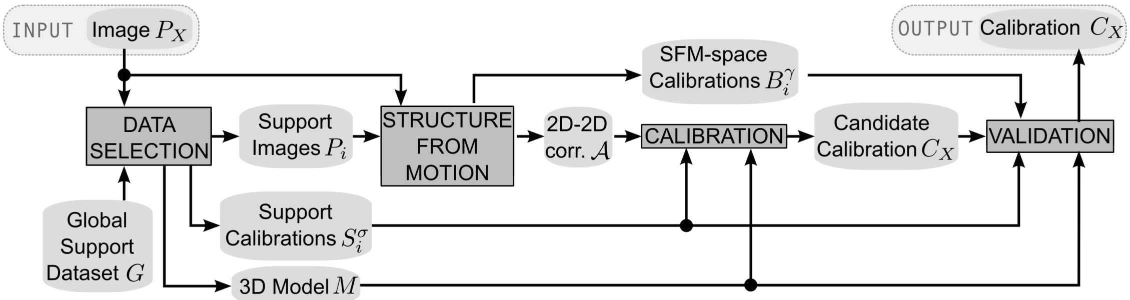

INPUT
Image P_X
OUTPUT
Calibration C_X
SFM-space Calibrations B_i^γ
DATA SELECTION
Support Images P_i
STRUCTURE FROM MOTION
2D-2D corr. $\mathcal{A}$
CALIBRATION
Candidate Calibration C_X
VALIDATION
Global Support Dataset G
Support Calibrations S_i^σ
3D Model M

Figure 5.1: Overview of the algorithm and data flow.

5.2 Geometry-aware automatic image localization

Our system deals with two specific requirements: the localization has to be automatic and accurate enough to allow correct superimposition of the input image on the 3D model for presentation purposes to tourists. There are no strict time constraints.

Our solution combines a state-of-the-art image retrieval system, a Structure from Motion algorithm and solutions coming from 2D/3D registration to recast the problem in a large-scale 2D/3D registration problem.

In the following the camera model is defined by 7 parameters: position, orientation, and the focal length. For the intrinsic parameters, we assume that the skew factor is zero, the principal point is the center of the images, and the scale factors are assumed to be known from the resolution and the CCD dimensions.

5.2.1 Overview

The basic idea of the algorithm is to use an efficient image retrieval system in order to select relevant and local information from the support data (implicitly obtaining a rough approximation of the location), then use such support data to calibrate the camera. Subsequently, the obtained calibration is validated in an unsupervised way to guarantee high accuracy. The data flow is shown in Figure 5.1.

The input is an image, P_X, that needs to be localized and calibrated. The *Data Selection* stage takes advantage of a *global support dataset*, G. This dataset contains a set of high-resolution 3D models (one for each location of interest), a set of images registered on the respective 3D model called *support images* and the corresponding camera parameters.

A retrieval image system is used to obtain a local subset of G composed by the k images P_i which are the most similar to P_X. The corresponding support calibration parameters S_i^σ and the 3D model M of the location of interest are also extracted.

The *Structure From Motion* stage uses P_X and the support images P_i to perform a Structure from Motion algorithm to obtain 2D-2D image correspondences $\mathcal{A}$ and auxiliary camera calibrations B_i^γ. This is done in a coordinate reference system γ

which is generally different from the reference system of the support 3D model, σ.

Calibration uses the 2D-2D image correspondences (computed at the previous stage), the 3D model M and the support calibrations S_i^σ to calculate a candidate calibration C_X of the input image.

Finally, the *Validation* stage uses the auxiliary camera calibrations B_i^γ, the 3D model M and the support calibrations S_i^σ to validate C_X.

Creation of the global support dataset

Concerning the creation of G, for each location of interest a set of images that covers as much as possible the model surface is acquired through a photographic campaign. Then, Bundler Structure from Motion tool [309] is used to produce an initial camera calibration of the images for each location, including a corresponding point cloud.

Since the results lie on a 3D frame coordinate system that is generally different from the coordinates frame σ of the 3D model, we align the 3D points using Meshlab [64]. We obtain a similarity matrix Θ that brings the set of calibrated images (and the calibrated data) in the σ reference system. The user can remove low quality calibrated images or attempt to adjust slightly wrong calibrated images by launching the fine alignment registration algorithm implemented in Meshlab. If an area of interest is not covered more images can be added to the set; in this case the Bundler Structure from Motion tool has to be re-launched on the expanded dataset.

5.2.2 Data selection stage

In this stage, Amato and Falchi [5] image classifier is used to obtain the subset $\{P_i\}$ of the global support images. The subset is composed by the k images which are classified as the most similar to P_X. This ensures the scalability of the system since this image retrieval algorithm can work for ten of thousands of images in a very efficient way. The algorithm performs a kNN classification using local 2D features (SIFT [206]). We refer to the original publication for the details of the algorithm.

We retrieve a list of 15 similar images (P_i) that form the *local* support images. This list is further pruned from possible outliers. Thresholding is applied over the similarity metric returned in order to have support images that share at least a partial set of features with respect to P_X. Then, a voting scheme is used to retrieve the 3D model of the location from the set of the available ones. At this point, the list is pruned by removing those images that refer to a different 3D model from the one chosen by the voting scheme. If the final list is smaller than a minimal set, 3 images, the alignment fails and the image is rejected. The local support images just represent a rough approximation of the location of P_X, since they are associated to a portion of a specific 3D model M.

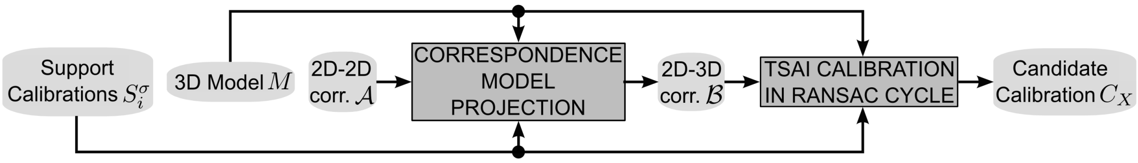

Support Calibrations S_i^σ
3D Model M
2D-2D corr. $\mathcal{A}$
CORRESPONDENCE MODEL PROJECTION
2D-3D corr. $\mathcal{B}$
TSAI CALIBRATION IN RANSAC CYCLE
Candidate Calibration C_X

Figure 5.2: Scheme of the *Calibration* stage.

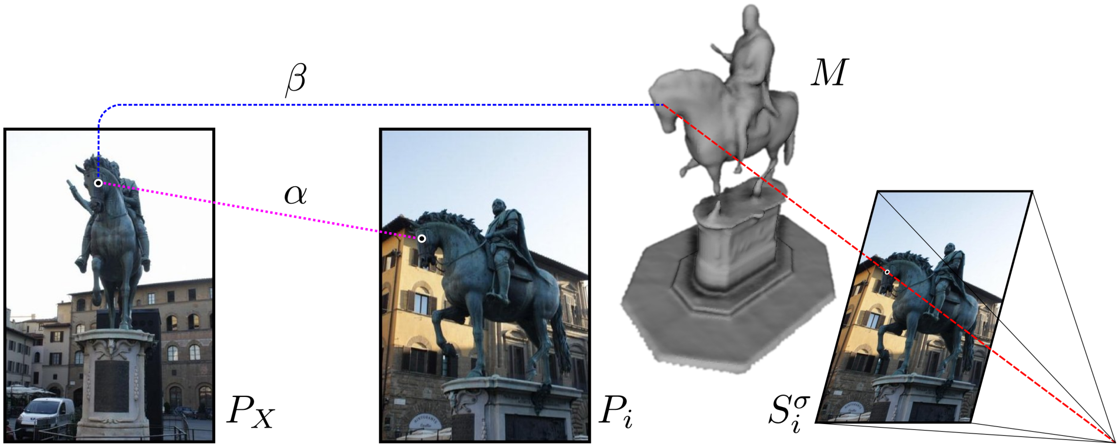


Figure 5.3: Correspondence projection scheme.

5.2.3 Structure from motion stage

The local support images together with P_X are given in input to Bundler [309] to obtain a set of camera calibrations B_i^γ, in a coordinate system γ, a set of 2D-2D correspondences $\mathcal{A}$ between salient features of the images, and a set of reconstructed 3D points in γ. $\mathcal{A}$ is employed in the *Calibration* stage, B_i^γ are used in the *Validation* stage.

5.2.4 Calibration stage

The *Calibration* stage follows the scheme in Figure 5.2. The goal is to obtain a candidate calibration C_X of P_X on the 3D model M. In order to compute it, we need a set of 2D-3D correspondences $\mathcal{B}$ that matches points on P_X with a set of surface points of the 3D model. These correspondences are not known in advance. Nevertheless, we can take advantage of the 2D-2D correspondences $\mathcal{A}$ between the local support images P_i and P_X, and the corresponding calibrations S_i^σ. These allow us to project features of P_i on the surface of M.

The procedure to build $\beta \in \mathcal{B}$ is shown in Figure 5.3: we project the feature point in P_i on the surface of M using the camera calibration S_i^σ and assign the 3D point with the corresponding 2D feature point in P_X. The projection is the intersection between the 3D surface and the ray connecting the feature point in the image plane with the point of view of the camera. If no intersection is found, no

2D-3D correspondence is generated.

During the construction of $\mathcal{B}$, there are many possible sources of error such as: false positives in $\mathcal{A}$, holes or incongruences between the model and the photographs (i.e. due to movable elements), small errors in camera parameters, etc. Even if multiple 2D-2D correspondences of the same visual feature in P_X are present, we keep all the possible 2D-3D correspondences. This is because our policy is to keep everything that is potentially correct and to deal with outliers in the following processing step. After obtaining the set of 2D-3D correspondences, we proceed with the effective calibration.

The calibration step follows a RANSAC [106] approach, that in each iteration selects a subset of $\mathcal{B}$, avoiding duplicates of the same 2D feature on P_X. Then, it computes a tentative calibration C'_X using the well-known Tsai [335] algorithm, and computes a projection error metric to select the "best" calibration.

This approach guarantees robustness with respect to outliers in $\mathcal{B}$. It also has a controlled processing time and avoids "local minima" problems. The RANSAC cycle is limited in time, because the processing time of each iteration is variable and depends on the number of correspondences. The time limit is set to one minute, but we enforce to have between 250 to 1000 iterations.

In each iteration, we randomly sample a constant amount of 20 2D-3D correspondences $\beta \in \mathcal{B}$ that are used for the calibration with the Tsai algorithm. Tsai calibration works with a minimal amount of 9 correspondences, but from our experience we found that 20 correspondences are preferable to obtain good results in presence of noisy data. Although this is apparently in contrast with the RANSAC philosophy of using minimal data sets, it works well in the practical case. This is because Tsai is robust to outliers, and it is de facto just a change on where to have the computational cost; i.e. we have more cost on the single iterations than on the external cycle.

After computing the candidate calibration C'_X, we measure its quality. For each $\beta \in \mathcal{B}$, we project its 2D point on the model surface using C'_X, obtaining the 3D point ρ. If the projection misses the model surface or if the distance of ρ from the 3D point in β exceeds a robust threshold we declare a miss, otherwise a success. C'_X is chosen as the calibration candidate C_X if there is any success, the misses are less than 10% of the total, and the average of the distances in successes is the best one.

5.3 Validation stage

The idea beyond our validation algorithm is to check the consistency between the estimated calibration C_X and the calibration parameters provided by the image-based reconstruction done with Bundler (see Section 5.2.3). Measuring the difference between two camera parameters set is not trivial. We decide to compare what the two cameras are "seeing" in the scene. To calculate such consistency measure, we do the following two steps:

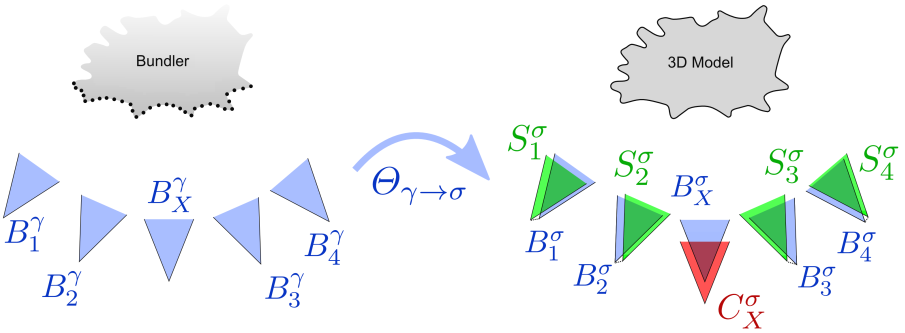


Figure 5.4: Least Square Mapping scheme.

1. We take the calibration B_X^γ given by the image-based reconstruction for image P_X and we map it in our coordinate frame σ, obtaining B_X^σ.

2. We measure how differently B_X^σ and C_X^σ view the same scene comparing two depth maps generated from these data.

To obtain B_X^σ, we exploit relationships between the support calibrations S_i^σ and the calibrations B_i^γ computed through Bundler; see Figure 5.4. The set of cameras has similar geometrical relationships in the camera positions, but differences in estimation are generated, e.g. due to the focal length/view direction ambiguity.

The estimation of the similarity matrix to obtain B_X^σ is performed following a RANSAC approach in order to account for outliers. A subset of the calibrations obtained is selected. The difference in scale is adjusted using the bounding box of the two sets of cameras. Then, the Horn method [157] is applied to estimate a similarity matrix $\Theta_{\gamma\to\sigma}$ to transform the coordinate frame from γ to σ.

We evaluate the quality of the similarity matrix by applying it to all calibrations B_i^γ and measuring the Euclidean distances between the viewpoints with respect to S_i^γ. The most accurate $\Theta_{\gamma\to\sigma}$ is applied to B_X^γ.

In the second stage, we check the consistency of B_X^σ and C_X^σ, by doing an image-based comparison on two virtual range maps. We opt for this novel approach since small changes in camera parameters can lead to major differences in the framed area, e.g. due to obstacles.

We proceed by obtaining two low-resolution synthetic range maps R_1 and R_2 of the 3D model as seen by the two cameras obtained. Then, we measure two errors: the *XOR consistency* (E_{XOR}) of model occlusion versus the background, and the Sum of Squared Differences (SSD) of depth values (E_{SSD}) between R_1 and R_2. The values of R_1 and R_2 are normalized *together* in the $[0\ldots1]$ range. Background values are set to ∞.

The XOR consistency (Eq. 5.1) is the percent of pixels that in R_1 are background and in R_2 are not and vice-versa:

$$E_{XOR} = \frac{\sum_{\substack{0<x<w\\0<y<h}} RX(R_1, R_2, x, y)}{wh} \tag{5.1}$$

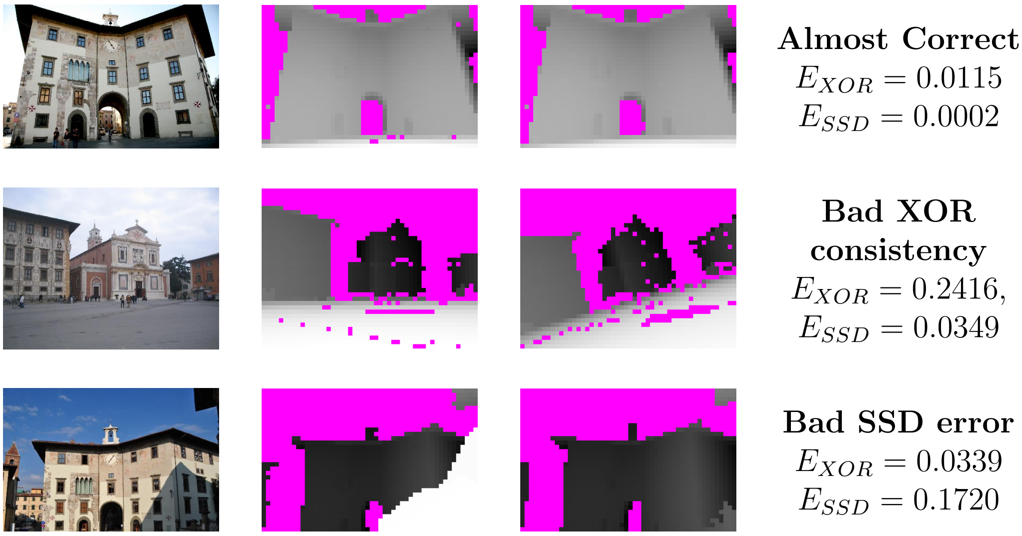


Figure 5.5: (Left) Input image. (Center) Range map obtained from C_X^σ. (Right) Range map obtained from B_X^σ.

where w and h are the size of the range maps and $RX(R_1, R_2, x, y)$ is defined as Eq. 5.2:

$$RX(R_1, R_2, x, y) = \begin{cases} 1.0 & \text{if } (R_1(x,y) = \infty) \oplus (R_2(x,y) = \infty) \\ 0.0 & \text{otherwise} \end{cases} \tag{5.2}$$

E_{XOR} essentially accounts for different positions and directions of the view.

The SSD error (Eq. 5.3) measures the dissimilarity in non-background areas:

$$E_{SSD} = \frac{\sum_{\substack{0<x<w \\ 0<y<h}} BH(R_1, R_2, x, y)(R_1(x,y) - R_2(x,y))^2}{\sum_{\substack{0<x<w \\ 0<y<h}} BH(R_1, R_2, x, y)} \tag{5.3}$$

where $BH(R_1, R_2, x, y)$ is the function of Eq. 5.4:

$$BH(R_1, R_2, x, y) = \begin{cases} 1.0 & \text{if } (R_1(x,y) \neq \infty) \wedge (R_2(x,y) \neq \infty) \\ 0.0 & \text{otherwise} \end{cases} \tag{5.4}$$

The E_{SSD} measure accounts for errors in position and focal length, that could lead to the different framing of objects which are near to the camera.
If $\sum_{\substack{0<x<w \\ 0<y<h}} BH(R_1, R_2, x, y) = 0$ there is a degenerated situation and E_{SSD} is artificially set to ∞.

Examples of the consistency measures are shown in Figure 5.5. The second and the third rows show two calibrations which are incorrect due to different reasons: in the first case, the XOR consistency results to be very high; in the second case, the problem is indicated by the value of the SSD error.

Table 5.1: Comparison of localization performances between our method, Li et al. [198], and Sattler et al. [278].

Tested method	average localization error (m)	# of registered images
Li et al [198]	18.3	94%
Sattler et al [278]	15.7	96%
GAIL (before validation)	3.9	59%
GAIL (after validation)	2.1	26%

5.4 Experimental results

In this section, we will describe and discuss the results of the experimental evaluation of both the full image localization algorithm and the validation step. The global support dataset is composed by images and 3D models for 2 locations: "Piazza Cavalieri" in Pisa (Italy) and "Piazza della Signoria" in Florence (Italy). The "Signoria" location is covered by 304 calibrated images while the "Cavalieri" location by 202 calibrated images. The corresponding 3D models (485k and 4083k faces respectively) have been obtained through ToF laser scanning, and prepared as explained in Section 5.2.1.

5.4.1 Comparison with previous work

In order to assess the performance of our system, we compared it with two recent state-of-the-art works in image localization [198, 278]. Both these systems were tested using the same "Dubrovnik" dataset [243], which is composed by 6844 images. The authors test their systems by extracting 800 images from the dataset, and try to localize them. Each test is repeated 10 times.

It was not possible to use the Dubrovnik dataset in our case, because no 3D model of the city is provided, However, we applied the same testing approach on our image datasets by attempting to re-align all the pre-calibrated images 10 times. Results are shown in Table 5.1.

Regarding the localization error, our method outperforms the others. The percentage of acceptance is lower due to the different goal, accurate calibration, of our approach with respect to the goal, localization, of previous work.

Figure 5.6 shows some examples of the calibrations obtained, we divide the calibration accuracy in: "high quality" (near pixel-perfect superimposition), "medium quality" (small misalignments are present), and "low quality" (severe misalignments with the 3D models or completely wrong result). Note that several of what we refer as "low quality" alignments could be accepted as correct by a typical localization system where only the position of the camera is important and not the orientation as well. Our goal force us to be more selective to ensure a satisfying navigation of the localized photographs. The thresholds set in the current system implementation, relative to the results here reported, allow for “high quality” calibration.

5.4.2 Result evaluation

In order to evaluate the performance of the system, the validation algorithm in particular, 568 input images were retrieved from Flickr, in order to cover many possible cases that the system has to face. These images have been manually inspected before testing, in order to have an “a priori” knowledge of which ones we expect to locate and which ones we expect to refuse. This classification is based only on the visual inspection.
We expect to calibrate images depicting:

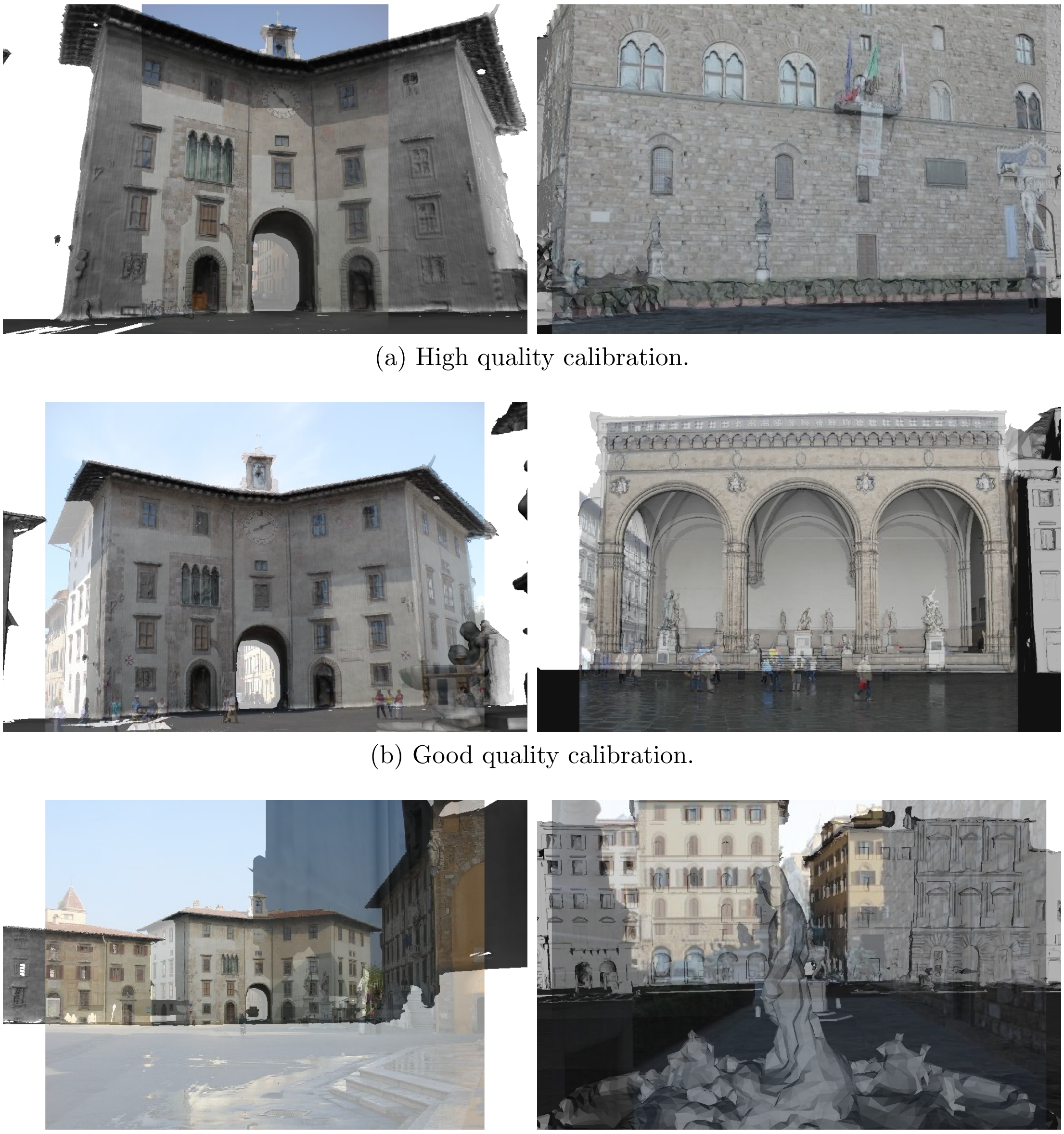
- façades of buildings, statues and other architectural elements present in the 3D model
- images with moderate clutter such as people, cars and similar occluders
- images with moderate amounts of reflexes and specularities, as rain puddles and windows

Some examples of images that the algorithm is expected to calibrate are shown in Figure 5.7.
We expect to refuse images depicting:

- areas predominantly not covered in the 3D model
- predominant clutter as closeups of people, movable objects, etc.
- very high zoom details of façades, statues and other architectural elements
- architectural discrepancies with the 3D model (e.g. scaffoldings)
- very poor illumination condition (e.g. shot in the night)
- photo-manipulated images (e.g. panoramas, composition, etc.)
- extremely blurred or unfocused images
- pictures taken with fisheyes, tilt-shift, bokeh, etc.

Some examples of images that the algorithm is expected to refuse are shown in Figure 5.8. Notice for example that image 5.8(c), is visually very similar to a picture of “Piazza della Signoria” in Florence but it is the picture of another plaza, is correctly discarded by the algorithm.

After this inspection, we expect to accept 319, 56%, of the 568 images and to refuse the remaining 249.
Of these 568 images:

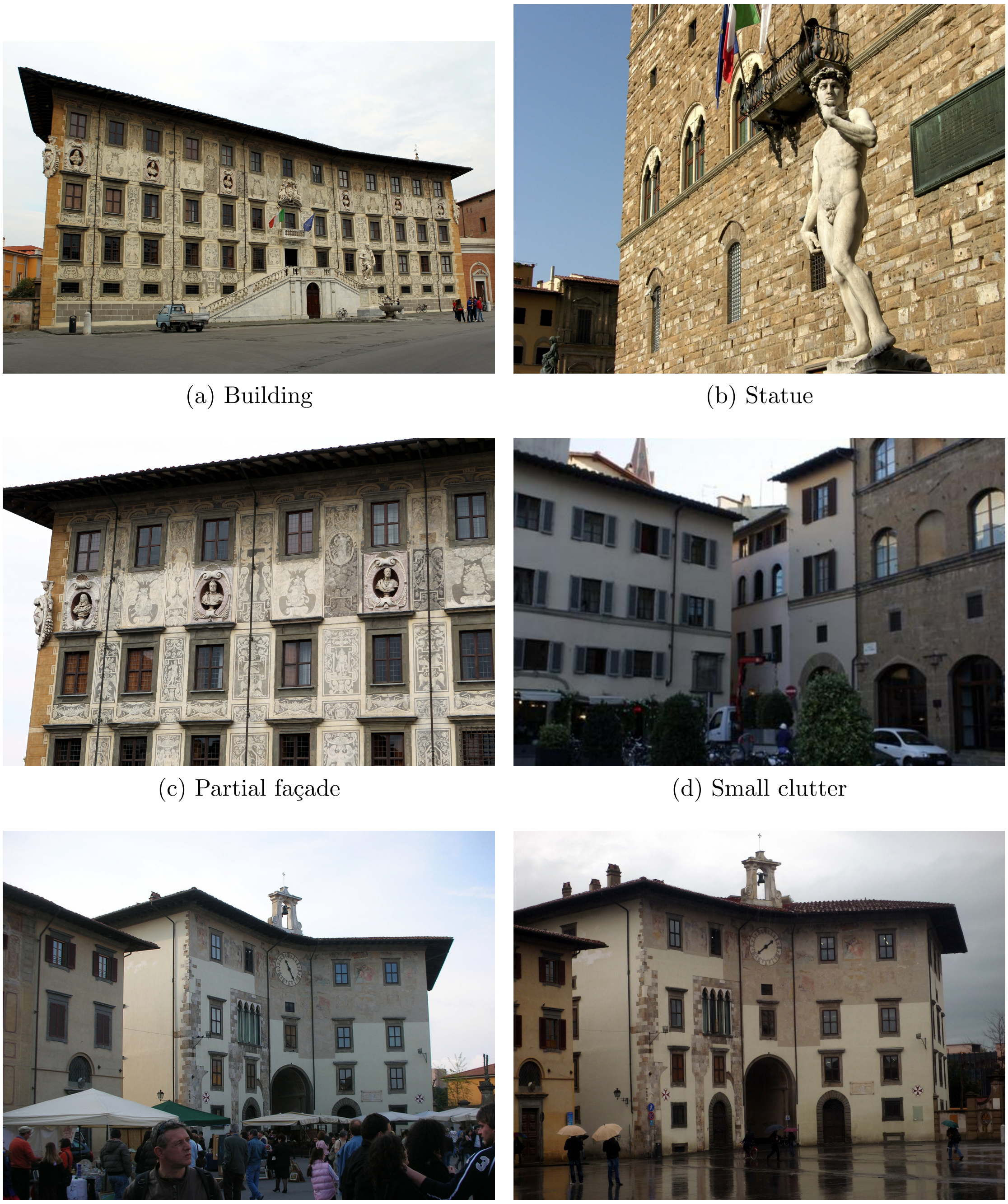


(a) High quality calibration.

(b) Good quality calibration.

(c) Bad quality calibration.

Figure 5.6: Calibration accuracy examples. We divide the calibration accuracy in "high quality" (near pixel-perfect superimposition), "good quality" (small misalignments are present), and "bad quality" (severe misalignments with the 3D models or completely wrong result). The thresholds set in the current system implementation, that are relative to the results reported in the Experimental Results Section, allow for "high quality" calibration.



(a) Building

(b) Statue

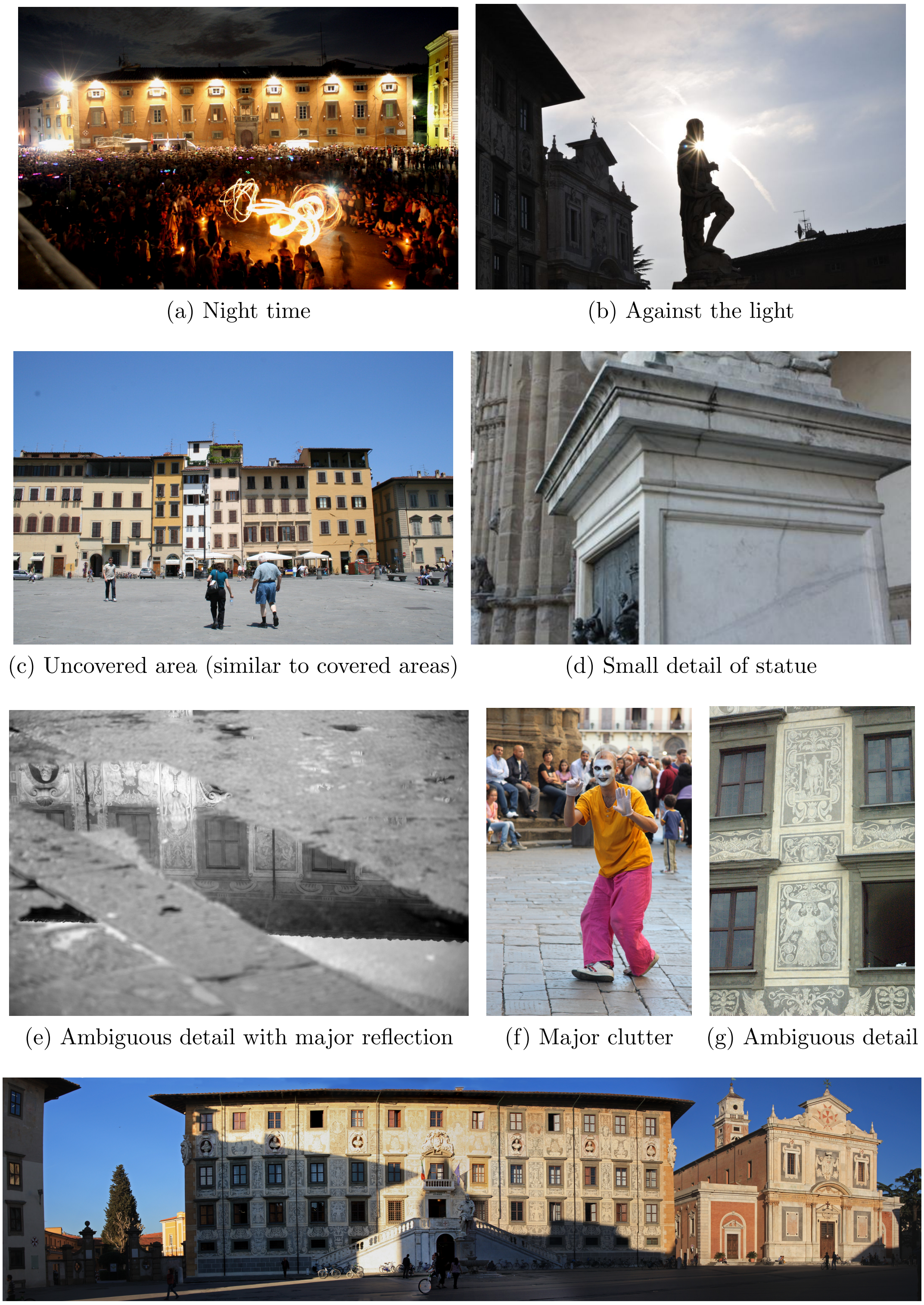
(c) Partial façade

(d) Small clutter

(e) Moderate clutter

(f) Moderate reflexes

Figure 5.7: Examples of images that we expect to locate.



(a) Night time

(b) Against the light

(c) Uncovered area (similar to covered areas)

(d) Small detail of statue

(e) Ambiguous detail with major reflection

(f) Major clutter

(g) Ambiguous detail

(h) Panoramic montage

Figure 5.8: Examples of images that we expect to refuse.

- 180, 31.7%, were rejected in the classification step
- 12, 2.1%, were rejected in the reconstruction step
- 146, 25.7%, were rejected in the calibration step.

This means that 230 images, 40.5%, were accepted by calibration. Among these:

- 119 (21.0% of total, 51.7% of selected) failed the *Validation* stage.
- 111 (19.5% of total, 48.3% of selected) were validated.

The thresholds used for the validation are $E_{XOR} \leq 0.15$ and $E_{SSD} \leq 0.05$.

The method proves to be very selective, since 38.7% of the images which were judged to be acceptable were discarded during the first three stages. On the other side, only 2 images, 0.4% of the total, were wrongly accepted. This is a key feature for a system which does not need any human-based validation of the results.

Moreover, the used datasets were not ideal, both in terms of input data (covering of the support images, quality of 3D model) and type of environment ("Piazza della Signoria" contains several statues, so that some images depict details which are difficult to match due to several occlusions).

We expect that the performance could be improved using a more complete (in terms of coverage) global support dataset.

5.4.3 Timing

Concerning the processing time of the different stages of the system, the time to retrieve the local support set is negligible, since the algorithm by Amato et al. is designed to deal with millions of images, and the global support set is usually composed by hundreds of images. Due to this fact, the time for finding similar images is practically instantaneous. The calibration stage, as previously stated, is limited to 1 minute (ensuring that a certain number of iterations is reached) in the current implementation, but further optimizations can be achieved to reduce this processing time. Finally, the validation stage is quite fast, in the order of tenths of ms on a average-end PC.

5.4.4 Discussion

The main advantage of the proposed system is that it can work in a completely automatic and unsupervised way producing very high accurate camera calibrations for urban context. This implies also a very accurate localization. The selection of images is very strict, in order to ensure the lowest possible rate of false positives.

This also gives the possibility to the system to "train" and increase the robustness, since the successfully calibrated images can be added to G in order to increase the performance of the system itself during its use. Moreover, a very high number

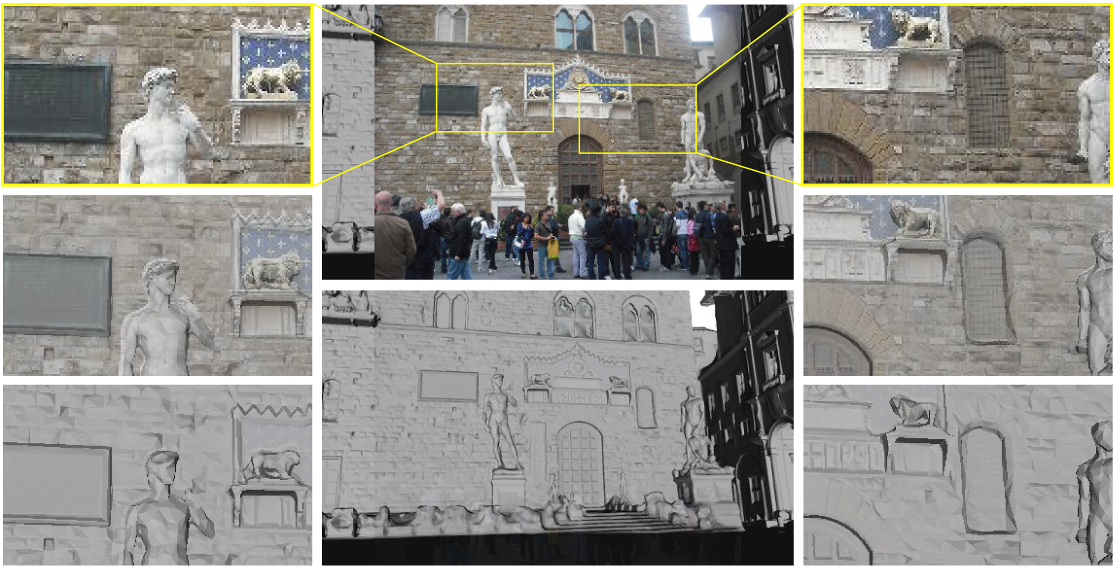


Figure 5.9: Results example. Center top: input image immersed in 3D. Center bottom: 3D model view from C_X^σ. Side columns: superimpositions of details.

of pre-calibrated images could be used, due to the scalability of the image retrieval algorithm employed.

The main limitation is related to the fact that the validation step discards a calibration more frequently for errors in B_X^σ than for errors in the *Calibration* stage C_X^σ. This means that the Bundler calibration is currently the weakest part of the system. More research in this direction could be of great interest, in order to have more validated results. Another limitation is that a 3D model of the scene is needed. Nevertheless, current multi-view stereo reconstruction techniques are probably able to provide an accurate enough reconstruction of the scene.

The system proves to be selective, but very accurate and robust. Thus, all the calibrated images could be directly used in a photo navigation system without the need of human validation. Figure 5.9 shows an example of an image that was perfectly aligned to a complex 3D scene, with objects of different sizes at different distances with respect to the point of view.

Chapter 6

Visualization of multi-view data

On the side of ***visualization of multi-view data****, we present PhotoCloud; a real-time client-server system for interactive exploration of large datasets comprising high-complexity 3D models and up to several thousand photographs calibrated over the 3D data. The system aims at generality and flexibility; so it is not tailored to any specific data acquisition process. PhotoCloud supports arbitrary photo collections and any 3D models that can be rendered in a depth-coherent way such as: point clouds, triangle soups, and indexed triangle meshes. It tolerates 2D-to-2D and 2D-to-3D misalignments. It provides scalable visualization of generic integrated 2D and 3D datasets, exploiting data duality. A set of effective 3D navigation controls, tightly integrated with innovative thumbnail bars, enhance the user navigation of the data. The scope of PhotoCloud is wide because the need to manage integrated 2D and 3D sampling arises in many domains: industrial plant inspection, city management, decision-support systems for crisis management, etc. A particularly important application context is Cultural Heritage, which often requires efficient, easy browsing of photograph collections, often referenced over complex 3D models. PhotoCloud effectively supports the exploration of virtual monuments, museums, archaeological sites, streets, plazas, and entire cities.*

6.1 Introduction

This chapter introduces *PhotoCloud*; a real-time system for interactive remote exploration of large multi-view data sets composed of high-complexity 3D models joint together with up to several thousands of photographs calibrated over the 3D dataset.

In last years, the diffusion of explicit 3D geometry in multi-view datas sets has increased. This is either due to image-based reconstruction techniques (seen in Sec. 2.1) or active 3D acquisition campaigns paired with intensive photographic samplings. Either way, the images and the model are reciprocally calibrated in the final mixed 3D/2D dataset. This means that pictures position and orientation (in the same space embedding the 3D models) are known.

Traditional texture mapping techniques allow to map photographic images onto 3D models, but they cannot provide a perfect solution for these kind of datasets for a number of reasons:

- *2D-3D incoherence:* the images represent information that does not exist in the 3D model (e.g. transient data, people, cars, scaffolding, etc.), or that is represented at a lower resolution, as small details are usually lost in the 3D acquisition;
- *2D-2D incoherence:* photos are taken in different conditions (e.g. time, light, geometry) and these differences are part of the richness of the data and should not be canceled by the blending approach used to produce the final texture;
- *data density:* the amount of photographic data is massive, often in the order of gigapixels, and with too high complexity to be managed with classical texturing approaches.

Conversely, mixed datasets of this nature require a presentation approach that cannot be based on just the 2D medium or the 3D medium. On one hand, images complete and integrate the information represented in the 3D model. 2D images enrich the 3D data with context, for example showing transient or time-related details, like surface colors at different times of the day, with clues about unmodeled features like trees or population, and also they are usually available at higher resolution than the 3D sampling. On the other hand, a 3D model is more flexible; it is not being tied to any point of view or lighting condition. Furthermore, it can be navigated with continuity using intuitive spatial metaphors, whereas directly browsing a very large image collection is notoriously an awkward, time consuming task, even when aided by additional mechanisms like tagging or content-based search mechanisms. In this sense, the 3D model can serve as a useful auxiliary guide for image-browsing.

Our main application context is the management of Cultural Heritage data, where the final user wants to access and explore massive photographic campaigns referenced over complex 3D models. The scope of PhotoCloud is not limited to this context, since the need of managing integrated 2D and 3D sampling is common to many other domains, for example industrial plant inspection, cadastral city management tasks or system to support the decision process in crisis management.

Several problems of different nature arise in the context described above.

Exploring a collection of thousands or more images poses a challenge by itself. Typically, image browser mechanisms do not scale up in the number of images. For example, tasks like identifying images featuring a sought view, or even plain visualizing the dataset as a whole quickly become unfeasible when the number of image increases over a few tens.

Interactive web-based navigation of the 3D model itself requires *real-time rendering* of highly complex geometric data. It also requires effective user interfaces to let the user choose appropriate points of view.

In between these two tasks, the *joint visualization* of 3D and 2D data requires specific solutions.

Similarly, a *joint interface* is required to address the two tasks of 2D image browsing and 3D model navigation, taking advantage of their dual nature.

Another set of problem arises from the need to access huge amounts of data from the client side in a *distributed environment*, avoiding excessive lagging times. The size of the dataset keeps being an issue even after it reached the client side, as it can be too large to fit in GPU RAM or even central RAM.

The above issues add to the ones which have to be overcome in *preprocessing*, in order to get the dataset ready. In PhotoCloud, this phase includes preprocessing of the 3D model (construction of the multiresolution structures, simplification, denoising, etc), calibration of the images, computation of semantic distances and linear ordering among the images.

6.1.1 Contribution

PhotoCloud is an integrated and interactive system which assembles several interconnected modules and techniques to successfully support its functionalities. While each of the PhotoCloud modules can be seen as an adaptation or an improvement of techniques already present in literature, we claim that the overall combination of these components results in an original integrated system which effectively addresses the intended problems in an unprecedented way.

PhotoCloud represents a tradeoff among different datasets peculiarities, provides a scalable system, and proposes to integrate thumbnail-bars to enhance user navigation over the data. PhotoCloud's support for the full integration with 3D models extends significantly the navigation experience. Differently from Street Slide [183], PhotoCloud uses a navigation approach which adapts both to bubble-like visits, and to broader views and movements, in that it is meant to be a more general purpose photo-navigator than the former.

PhotoCloud follows the overall philosophy introduced by PhotoTourism [309], but it improves over the latter in many ways such as: a more flexible management of the image thumbnail-bar, increased 3D data flexibility (e.g. the use of high quality 3D model), enhanced visualization and navigation features that fully exploit the underlying 3D dataset.

In PhotoCloud, highly detailed multiresolution 3D model and photographs are combined. The latter ones are projected on the geometry only if their view position differs from the current view position within a small threshold. Thus, for farther view interpolations, the need of ambient point clouds is avoided.

Our system focuses on the presence of an high-quality 3D model, and we build a user experience finalized to both provide an accurate perception of the shape of the 3D model even without the color information allowing a free navigation on it, and provide the possibility of projecting the color information over the model. On the other hand, PhotoTourism and PhotoSynth are focused over the idea of navigating a set of photos, and the 3D information is often scarce and used only as a way of "connecting the images". The image centered navigation approach has not been put aside in PhotoCloud, offering an advanced tile bar to present and navigate the photographic collections in the 3D context.

6.2 System overview

PhotoCloud supports the browsing of a digital image collection I in conjunction with the navigation and rendering of a 3D digital scene M. Both I and M can be large in size, respectively in terms of number and resolution of images (i.e. up to several gigapixels) and geometric complexity (i.e. tens of million 3D points or triangles).

We assume that both I and M have been acquired and processed in a preliminary phase including their reciprocal calibration, de-noising, cleaning, etc. Figure 6.1 shows the PhotoCloud pipeline. First, in order to be efficiently used in PhotoCloud, I and M have to undergo a specific pre-processing phase. M can come in any form, as long as it can be rendered in a depth-coherent way. Specifically, the system supports point clouds, triangle soups, and indexed triangle meshes of many formats. We do not expect M and I to be perfectly aligned or even reciprocally fully consistent. As mentioned, images in I can, and usually do, feature details and entire objects absent in M (e.g., in an outdoor scene, they can include people, cars, trees, or depict buildings as they used to be in the past). Likewise M can describe objects or details not visible in any image.

The client application window shown in Fig. 6.2 is divided in the 3D area, which is the main part on the top, and the thumbnail area, which is a smaller part at the bottom. The user can resize the two areas by dragging the boundary separating them.

The 3D area shows, in a integrated way: a 3D rendering of M, the single currently "selected" image $i_s \in I$ (if there is one), and a few other images from I; which are the more pertinent ones in that given moment of the navigation. Images other than the selected one are shown in the 3D area as simple 3D glyphs termed "framelets". Framelets serve to signal the presence of relevant images in the context of the 3D navigation, to roughly indicate their contents, and as interface mechanism for browsing through images.

In the thumbnail area, images in I are represented by thumbnails, arranged in a focus-and-context thumbnail-bar, designed to scale well with the number of images in I. In order to cluster thumbnails and arrange them into proper 2D layouts, the thumbnail-bar employs precomputed image-to-image semantic distances and linear orderings in I . Thumbnails away from the current focus are more clustered and shown smaller. During the navigation, thumbnail layout and clustering are dynamically changed accordingly, without breaking temporal coherence.
PhotoCloud functionalities can be conceived as follows:

- *PhotoCloud as a scalable 2D image browser for large image datasets.* Tradi-

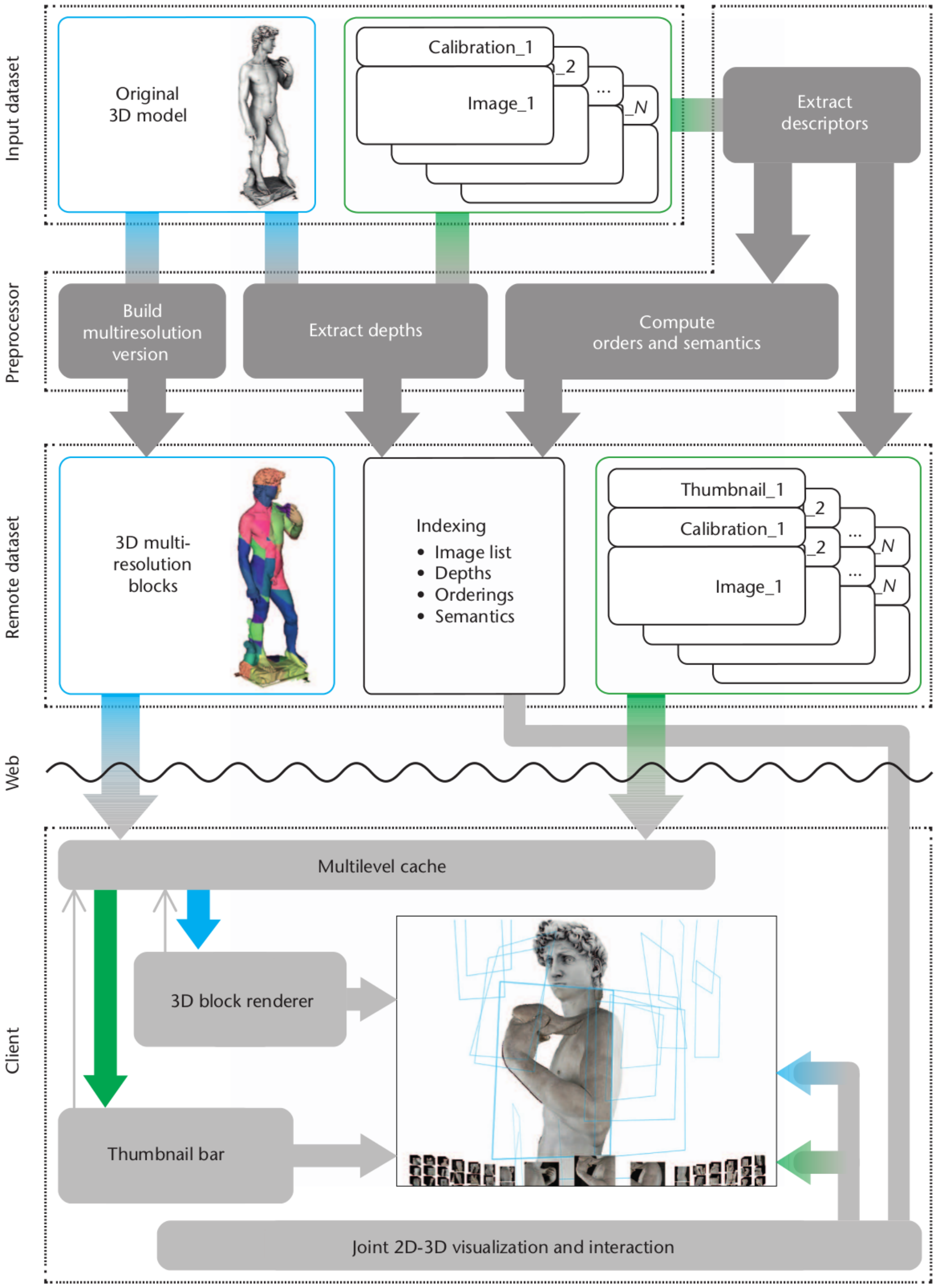


Figure 6.1: PhotoCloud overview. The input comprises a 3D model and a calibrated image set. The preprocessor converts the 3D model to a multiresolution format and computes image thumbnails, the average depth, the ordering, and the semantic distance. The system then creates an index file containing references to the model and images, associating them with their depth, order, and semantic information. A remote server stores the preprocessed data. The client downloads data through a unified cache system. Blue arrows represent the flow of 3D data; green arrows represent 2D data.

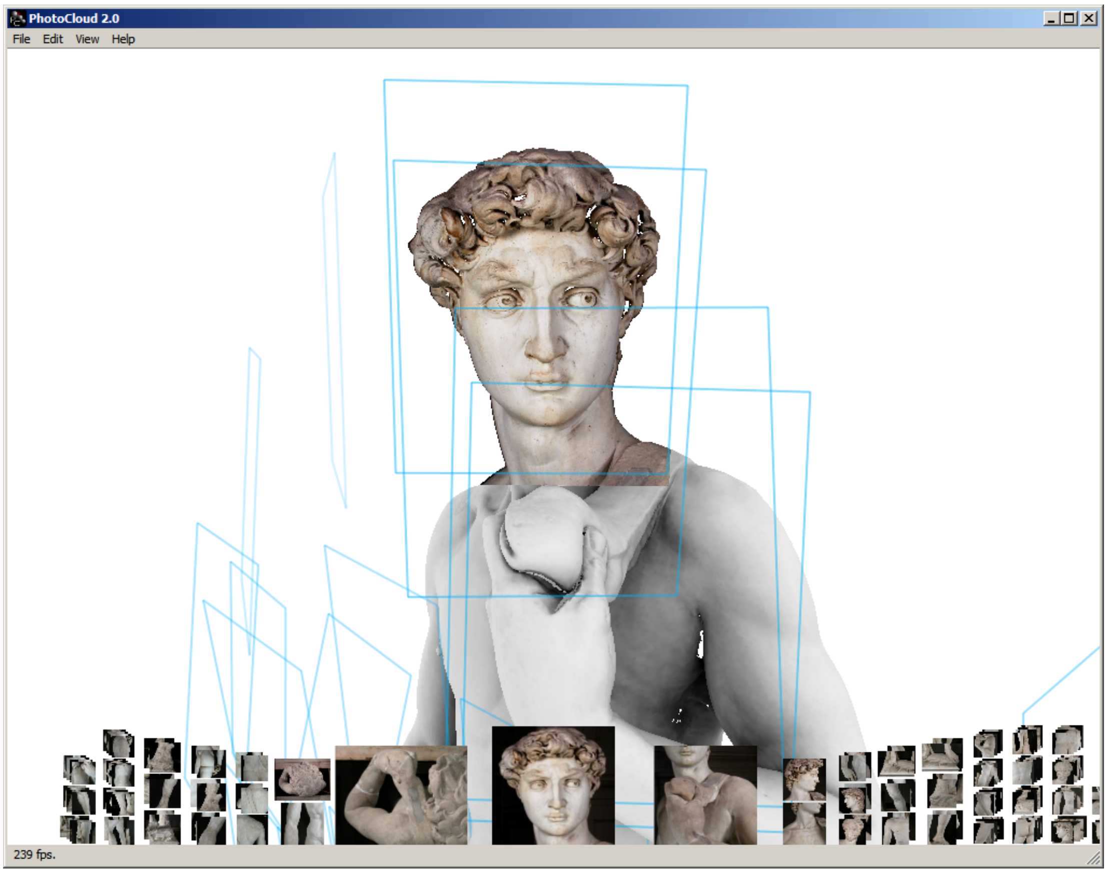


Figure 6.2: PhotoCloud main window integrates 3D and 2D representations of the dataset. All images are rendered in the bottom thumbnail-bar, which constitutes the images visual-content browser. The currently selected image is also rendered on the 3D area overlapped on the model, while framelets represent images with similar views.

tional 2D thumbnail-based image browsers usually show the currently selected image at full screen resolution (and other images as thumbnails). Similarly, in PhotoCloud, when an image is selected from the thumbnail-bar, that image is shown inside the 3D area by projective texturing mechanism, and simultaneously the 3D viewpoint is moved to coincide with the camera shot of the selected photograph. The final effect is equivalent to show the selected image at full screen resolution in 2D; even though in PhotoCloud this blends seamlessly with the 3D rendering of M during the rest of the navigation;

- *PhotoCloud as a scalable virtual 3D scene navigator.* PhotoCloud embeds a 3D mechanism for navigating scenes and, thanks to the adoption of an advanced GPU-friendly multi-resolution schema, it is capable of streaming and rendering very high-resolution models in real time;

- *PhotoCloud as a scalable integrated 2D/3D navigator.* The most interesting opportunity arises from the conjunct use of the two interface approaches, where each of the two serves as a powerful aid to the other. Hence, in PhotoCloud, the array of interfacing mechanisms triggered from the 3D area also affects the thumbnail area, and vice-versa. This holds for all user actions: selecting images, determining view positions, previewing images, etc.

PhotoCloud is designed to run in a web-based environment, where the data $(M + I)$ is kept in a remote server and accessed through a client. To this end, an efficient multi-layer, GPU-friendly cache system has been designed, managing both the image set I and the nodes of the multiresolution structure for M. This also serves as a way to efficiently handle datasets which exceed the capacity of the client's GPU (or even central) RAM.

6.3 3D visualization

As shown in Fig. 6.2, the PhotoCloud window is subdivided in two partially overlapping parts: in the main area above, covering the most part of the screen, the 3D model is shown rendered from the current point of view. In the bottom part, the 2D image collection is visible by showing the associated thumbnails in the thumbnail-bar. In the 3D model area, framelets are rendered too, providing a bridge between 3D and 2D interfaces. This feature is also exploited to enhance user navigation inside these large datasets.

6.3.1 Rendering the 3D model

The module performing the rendering of the 3D model is largely independent from the rest of the application. The 3D rendering is performed according to the nature of the 3D data: point-clouds are splatted (see Fig. 6.3), as in Rusinkiewicz and

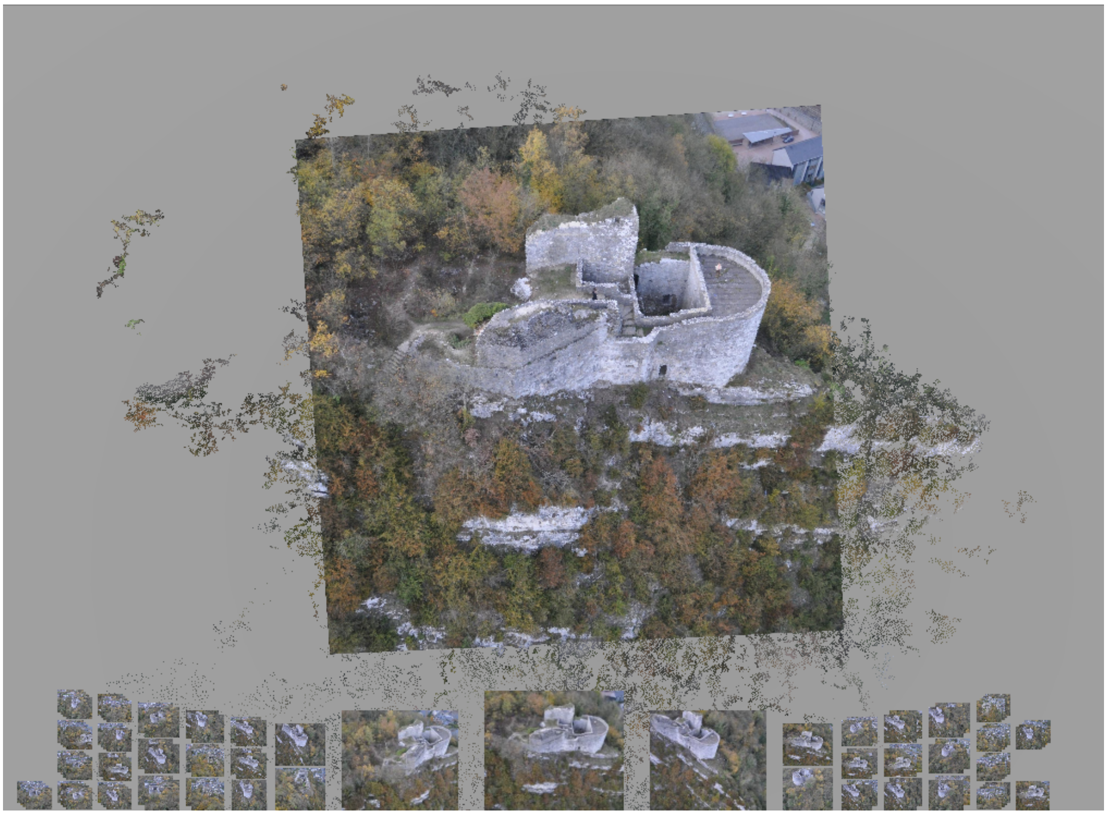


Figure 6.3: Point-cloud splatting. This figure shows an aerial view of a castle, geometrically represented by a dense point-cloud efficiently and rendered with the adopted multiresolution technique, while an image is rendered on it. Their combination effectively enhances the dataset presentation.

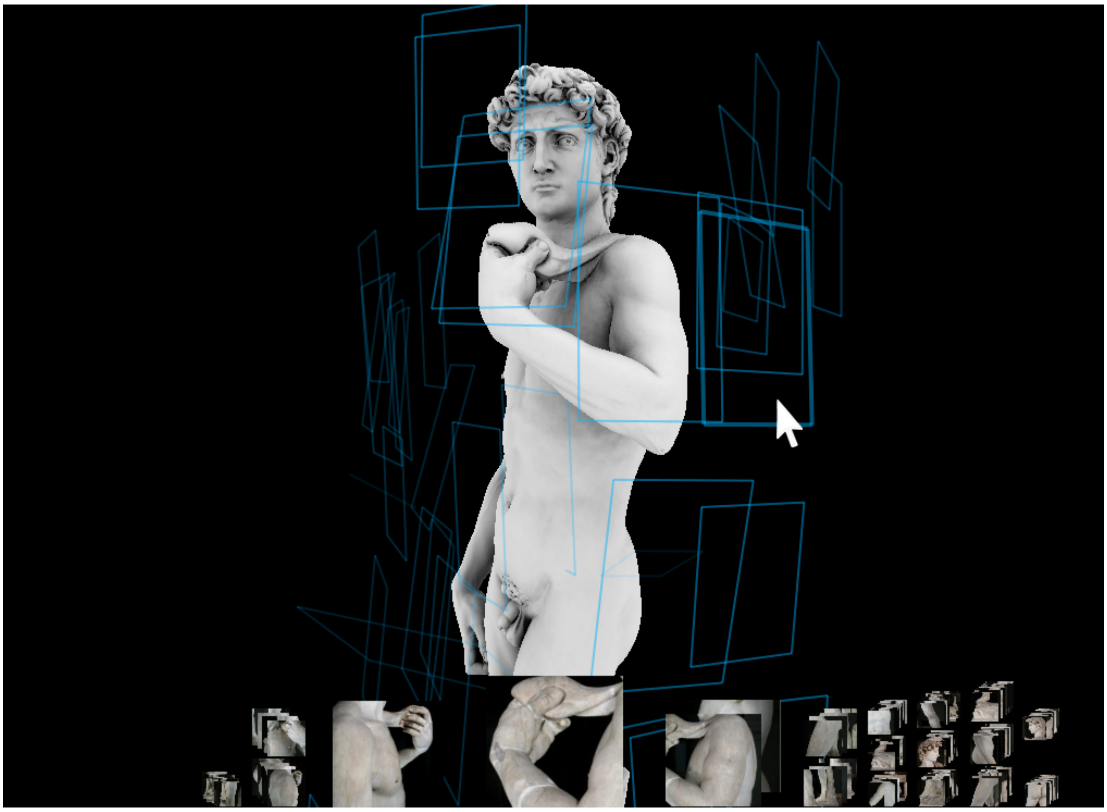


Figure 6.4: Framelets rendering. In the 3D model rendering area, those images having similar views to the current one at the model (i.e. front-facing the current view and forming angles of up to 90 degrees with the current view direction) are represented as framelets: texture-less colored rectangles. In this figure, they are shown in blue and they are more opaque for similar view directions, while tend to disappear for perpendicular ones. The framelet under the pointer is emphasized by a slightly thicker line and full-opacity.

Levoy. [274], triangle meshes and triangle soups are sent to the GPU as VBO and rasterized [65]. PhotoCloud also supports attributes like color (that also helps to encode a precomputed ambient occlusion factor in our datasets) and normals, defined per vertex on the models. Common effects, like depth cueing by fog or dynamic relighting, are added as needed by the application context.

The problem of rendering a highly complex 3D data structure, which is remotely stored and whose geometrical complexity would easily surpass the triangle-rate of the graphic card, is tackled resorting to a state-of-the-art multi-resolution data structure. As new LODs are loaded into GPU memory, the rendering is updated.

6.3.2 Framelets

Images other than the selected one are shown in the 3D area as “framelets”. A framelet is a simple 3D glyph representing a picture as a 3D outlined semitransparent colored rectangle, with the same aspect ratio as the represented image (see Fig. 6.4). The rectangle is defined as the section of the view frustum pyramid of the corresponding shot, cut at a distance from the camera roughly corresponding to the (precomputed) depth of the objects featured in the image.

In this way, the 3D position, orientation and size of the framelet rectangle reveal to some extent the intrinsic and extrinsic parameters of the shot, which in turn allow the user to predict their content by looking at the spatial relationship of the framelet and of the 3D objects. Framelets, which are unrelated to the current view (i.e. the ones which are relative to the image shot toward the current view position), are hidden to reduce screen cluttering and improve visual perception.

Framelets are simply drawn as wireframe rectangles. They are visible only when seen frontally (and are hidden when seen from behind). As an additional visual hint, framelets seen from the side are drawn progressively more transparent; i.e. the opacity of a framelet goes with the cosine of the angle between the current view direction and the direction of the corresponding shot.

6.3.3 Rendering images on the 3D model

Sometime the geometry has to borrow colors from the image content. The large number of images typically available in a mixed 2D/3D dataset potentially could allow for the photographic color information to be statically transferred and baked on the 3D model vertexes in order to have a viewpoint-independent colorization of the scene.

An issue of following such an approach, is that the image set might not be static, i.e. new calibrated images can always be added in the dataset, and thus we need a dynamic way to transfer color to the 3D model at rendering time. In the simplest dynamic case, a single image can be used to color the image with projective texturing [285], which consists in shooting an image RGB from its view point $view_{PT}$ onto the geometry. The multiple case can be resolved using techniques which blend RGB information from multiple source images [83]. In these cases, images are often preprocessed in order to extract albedo (e.g. deshade and highlight removal, as in Callieri et al. [55]).

We experimented with this approach, by implementing new techniques such as GPU-based multiple projective texturing, that can blend. [55]) efficiently multiple textures using a deferred shading [85] approach. In this approach, the model geometry is rendered only one time and different GLSL fragment shader passes are done to accumulate the color contributions of the various projective textures. A final fragment shader pass blends the color contribution from single calibrated images. This approach works really well when there are guarantees on the quality of the calibrations and the consistency of the photographic contents.

In the more general case, these two assumptions cannot be made. For example, images can depict the same object at different times of the day and at night, as well as from slightly different viewpoints, and even feature different temporary objects (e.g. bystanders, cars, etc). In all these scenarios, blending different images invariably leads to disturbing artifacts. Moreover, in the general case, we cannot assume data precision, not only in terms of controlled illumination at image capturing time, but also on the precision of the image calibrations.

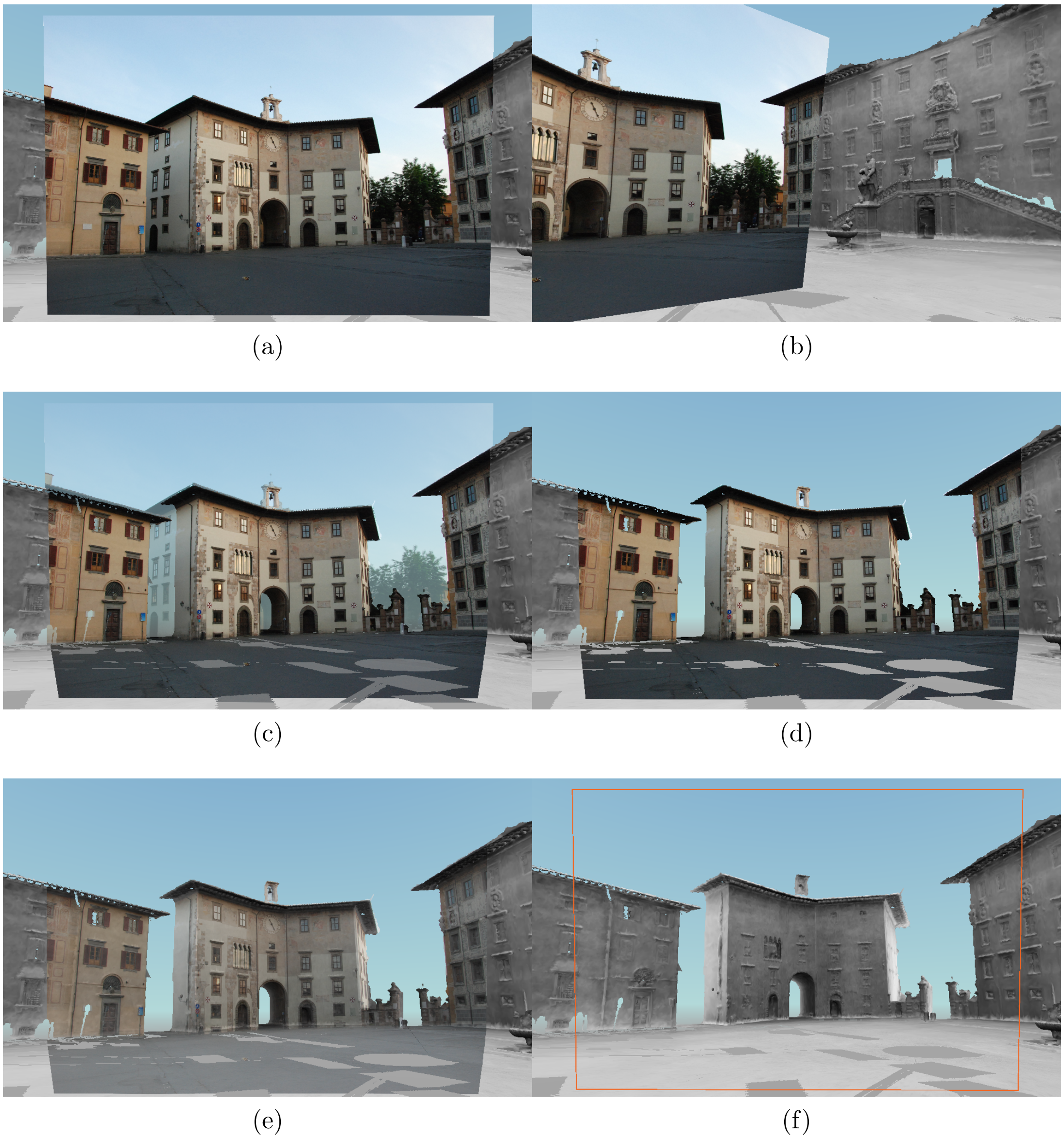
Finally, a goal of PhotoCloud is to support the easy visualization of multiple appearances of the depicted scene. This would need to explicitly define different subsets of the image collection that depicts these multiple appearances when the photographic contributions are merged.

In the light of all those considerations, we resort to simple projective texturing of *at most one image*, the currently selected image, to color the geometry for this kind of datasets.

This technique produces correct renderings whenever the model is viewed from $view_{PT}$, independently from the view direction. A skydome mesh, a large sphere encompassing the entire model and the viewer, is added on the background to fill the entire screen. In this way, the projective texture is projected over it in places not covered by the geometry; this allows also to paint the sky.

When the view position, direction, and field-of-view coincide with $view_{PT}$, the overall effect is indistinguishable from looking at the selected image at full screen, as common in traditional image browsers, see Fig.6.5(a). This avoids the need of different techniques (e.g. by rendering a textured quad on the foreground) for showing the selected image at full screen.

The advantage of the projective texturing approach is that it can also be used when the scene is viewed from other positions and directions. In particular, no artifacts occur when the view position matches that of $view_{PT}$, regardless of view direction and of any discrepancy between the 3D geometry and the image content. Hence, the view can be rotate for arbitrary angles, still obtaining consistent geometry coloring; a user cannot distinguish the projected photograph from a hypothetic perfect 3D model. For slightly different view positions, the result still looks consistent, but the more different the view position is, the more the 3D-2D discrepancies generate evident projection artifacts. Thus, as the distance between the viewpoint and pos_{PT} increases, we progressively fade-out the RGB color of the texture: $fade = K \times abs(pos_{CV} - pos_{PT})$, where pos_{CV} is the position of the current view, and K is the fading speed. As projection artifacts depend on geometric discrepancies, and higher discrepancies are produced for the skydome (whose depth is assumed to be very large), the value of K is differently set for the model (low) and the skydome (high). The net effect is that near $view_{PT}$ the sky and the background items depicted in the image disappear more rapidly (see Fig. 6.5).



(a) (b)

(c) (d)

(e) (f)

Figure 6.5: The adopted texture projection technique. All screenshots refer to the same selected image projected on the 3D model plus an additional skydome background. Only the view position and orientation change. When the view exactly matches the image view (a), projecting the texture has the same effect of rendering a textured quad above the model. The projection is still consistent for arbitrary view directions (b), but translations determine mismatches between the image and the model depending on the distance of the model from the viewpoint. To hide projection artifacts, the image is gradually faded out for slightly view translations, more rapidly at the background (c, d). When the view-position discrepancy increases the texture projection is progressively disabled (e, f).

6.4 Interaction mechanisms

One basic way to interact with Photocloud is through the clustered thumbnail-bar featured at the bottom, using any of the basic interaction mechanisms of that widget (including selection of an image, reordering of images, scrolling, previewing an image, etc). Likewise, acting on the main part of window (on the top), where the 3D model is featured, Photocloud offers a set of standard 3D navigation mechanisms, detailed here for completeness. These two modules also affect each other so that the user always has a coherent view of the dataset.

6.4.1 Navigation of the 3D scene

Moving around the 3D scene, the set and the opacity of the visible framelets is updated. In this way, the user can more easily find images with similar viewpoints. Projective texturing is enabled whenever the user is passing near one of them. When the user selects a framelet, the view rapidly flies to the associated view, also enabling texture projection. During the navigation, the user can always break any transition and choose a different point of view with the other controls.

Navigation controls can be customized depending on the nature of the dataset. In any case, the pointer (e.g. mouse or touchscreen) and key interface (e.g. keyboard or pad) are used to let the user determine the 6 degree of freedom of view position and orientation, plus focal length. For datasets featuring virtual environments like a square or an open ground, Photocloud adopts a freely moving avatar metaphor. The mouse controls the view direction (up direction being constrained to point away from the ground), and the current point of view can be moved in the horizontal plane via a keyboard interface (also known as WASD navigation in game-oriented communities, referring to the typical First Person Shooter interface). Another key controls the field of view, and the mouse wheel controls altitude. For datasets featuring a single object of interest like a statue, a trackball interface is adopted, where mouse drags let the user move over the surface of a two-manifold ellipsoid around the object under inspection. Every point p over the two-manifold controls both view position and orientation (pointing along the normal direction at p), even if the latter can be temporarily overridden by means of right-button mouse drags.

6.4.2 3D to 2D

When the pointer moves over a framelet, the corresponding thumbnail preview is enabled inside the thumbnail-bar. Furthermore, every time a framelet is selected or a viewpoint change enables a different texture for projection on the geometry, a focus change is automatically triggered in the thumbnail-bar. This grants that the 2D bar is always coherent with the current 3D view.

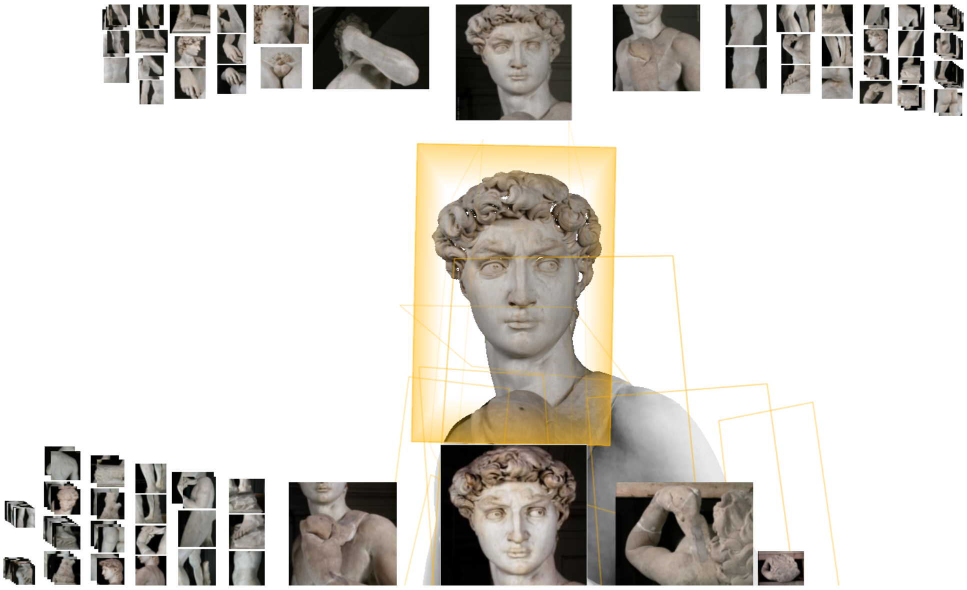


Figure 6.6: An example of advanced navigation using two thumbnail-bars.

6.4.3 2D to 3D

In the thumbnail-bar, moving the pointer over a thumbnail highlights the corresponding framelet (if visible), while selecting a thumbnail triggers a view change in the 3D viewer through a soft transition. These connections keep the 3D view coherent with the thumbnail-bar focus. Instead, thumbnail scrolling and dragging facilities, which determine focus changes, are not connected to the 3D view, as in this case the user is meant to look for a specific image by analyzing the thumbnails visual content. Sequences of scrolls and drags usually end with a thumbnail selection, thus triggering the 3D view.

6.4.4 Advanced exploration modalities

The proposed framework is flexible enough to allow for more advanced data navigation modalities, like time-based ones. Figure 6.6 shows such an example, in which the user can switch between two different images sets, relative to the statue appearance before and after a restoration. These two image sets are presented using two separated thumbnail-bars. This allows the user to jump freely and seamlessly between visualization of the statue before the restoration and after it during the 3D navigation.

6.5 Multiresolution 3D data structure

Rendering of 3D data is based on the Nexus library [343], which allows realtime multiresolution visualization of massive 3D meshes and point clouds. The construction process adopted in Nexus is based on a revised approach inspired by the works by Cignoni et al. [66], and Gobbetti and Marton [127]. The original data is split into blocks at different resolution that can be assembled in different combinations to produce the full model, adapting the resolution of the geometry to the distance from the view-point in order to keep the primitive count as low as possible. Since each block consists of several thousands triangles and is precomputed in a preprocessing step, assembling at rendering time the view-dependent representation is extremely fast and results in very low CPU load. Each block is optimized, cached in the GPU and rendered with a single CPU call for maximum performance. The rendering algorithm selects the best representation according to the rendering budget and the availability of the blocks, thus guaranteeing a minimum frame rate. The data structure is out-of-core and supports compression and streaming over http, thanks to its clustered nature. The geometry is organized into a bounding sphere hierarchy which easily allow for occlusion culling and collision detection. A detailed discussion of algorithms and data structures used in the Nexus library can be found in Chapter 3 of Ponchio's PhD thesis [259].

When the model is too large to fit into GPU memory, a priority based cache system, GCache [342], ensures best allocation of resources. The design of such a cache system is challenging, because it requires managing thousands of items allowing for very frequent priority updates and locking of items, and synchronization of different threads. To minimize the overhead due to the priority sorting, we adopt a double heap-based queue coupled with lazy updating of the queue, resulting in negligible CPU usage. A number of resources need to be locked each frame for rendering; this needs to be implemented using atomic integer operations instead of mutexes. Each cache (HTTP, disk, RAM, GPU) operates in its own thread allowing for blocking operations on files and sockets, and greatly simplifying the implementation.

6.6 Embedded image browser

The content-based image browsing mechanism adopted by Photocloud employs a focus-and-context thumbnail-bar, following Brivio et al. [47], which dynamically arranges image-thumbnails into clusters displayed as stacked piles on a small horizontal bar, see Figure. 6.7. Here we report its key properties.

The adopted thumbnail-bar assumes that some linear ordering is defined over the images, which will be exploited to ease navigation. The ordering can be chosen among several possibilities, and even dynamically switched among them within the GUI. Additional information about the semantic distance among each couple of

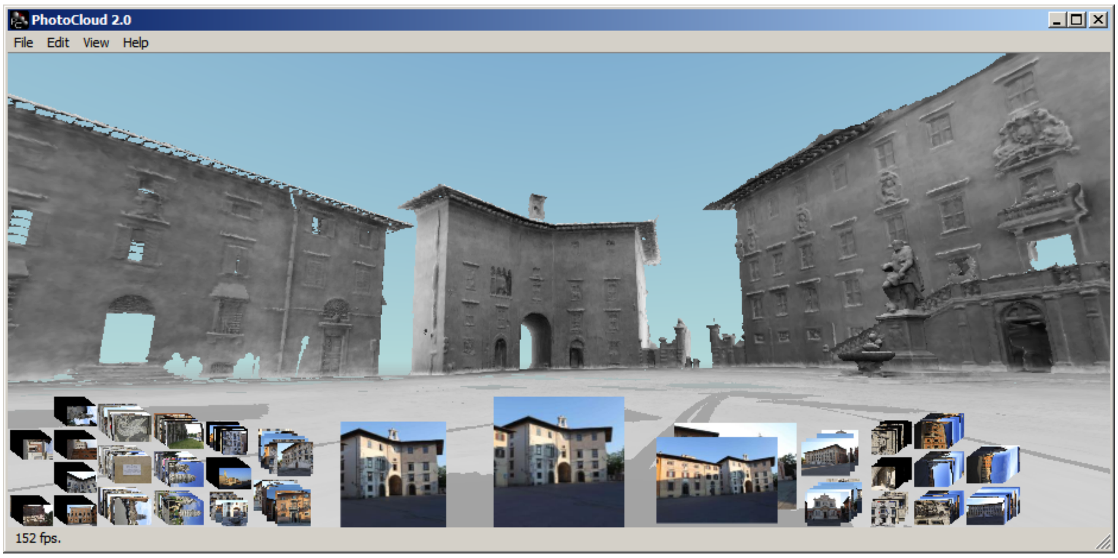


Figure 6.7: Thumbnail-bar close-up. This focus-and-context image browser arranges image thumbnails into stacked piles. The focus image is represented in the largest thumbnail in the middle of the bar. The others get clustered into piles of whose size depends on the distance from the focus. Images are assigned to a pile if they are near in a semantic domain. Scrolling, selecting, and previewing images complete the 2D navigation interface.

images can be used to better cluster thumbnails: closer images are clustered together, whereas more different ones are put in evidence in smaller piles.

The currently selected image is the "focus image", which is represented by the largest thumbnail in the bar and is located in the middle of it. Thumbnails nearer to either edge of the bar become progressively smaller and tend to be more clustered into stacked piles of increasing height. Each pile tends to contain the most semantically similar images within a threshold, which is very small near the focus but increases exponentially farther from it. Pile height is a visual indicator of the number of locally (in the ordering) similar thumbnails, all partially-covered and represented by the topmost one. In the bar layout, each pile is assigned a predetermined area, thus smaller piles reserve more pixels to each partially-covered thumbnail. This is useful during navigation, as the basic control consists in thumbnail selection.

When the user selects a thumbnail with the mouse pointer, that image becomes the new focus, and the entire layout is consequently rearranged into a new spatial configuration (affecting thumbnail sizes, positions, and clustering). The transition takes place as a smooth animation, and the new arrangement is selected in a way that also minimizes the movements across the screen. An alternative way to trigger a focus change is to drag any thumbnail across the bar, constraining its position with the pointer and changing the focus accordingly. Moreover, acting with the mouse wheel rapidly scrolls the thumbnails through the focus.

The thumbnail-bar offers also additional browsing mechanisms. Right-clicking on a thumbnail other than the focus triggers a preview above the thumbnail-bar, horizontally aligned with the thumbnail. The preview consists in a larger thumbnail of about the same size of the focus (i.e. the largest) thumbnail, offering a quick enlarged and non-occluded view of thumbnails away from the focus, and of the ones inside a cluster.

6.6.1 Exploiting images linear ordering

In the thumbnail-bar adopted in Photocloud, the image linear ordering is used to arrange the thumbnails in the layout (ordering along the x-axis of the bar reflects the given linear ordering of the images). This makes the browsing more intuitive, as given a thumbnail all thumbnails appearing at its left come before it in the linear ordering, while all the others come after it. Inside each stack, piles are also arrange in the same order, to minimize thumbnail movements during the bar updates.

Various linear orders can be imposed on the input image collection. A useful navigation order is the time-sequence of the shots, which emphasizes the timing information (i.e. older images whit respect to the focus one are placed on the left half of the bar). In our tests, we computed other orderings also on multi-dimensional domains, such as image color-distribution (i.e. based on color histogram), image spatial-color-layout, (i.e. encoded in a 4×4 down-sampled image), and image calibration (i.e. a 3D translation vector and a 2D rotation vector). We call the chosen domain D_O. Then, we define image distances as Euclidean distances for both the color-distribution and spatial-color-layout vectors. In the case of the image calibration domain, we compute translation and rotation Euclidean distances separately, and linearly combine them with a 0.5 coefficient.

To linearize D_O, we have to compute an Hamiltonian path. Due to the computational complexity of the problem, we adopted an heuristic to approximate the result and speed up the computation.

In any case, the ordering has to be precomputed off-line.

Photocloud is not aware of such pre-computation and only reads-in the cardinality of each image, which will suffice to constrain thumbnails horizontal position.

6.6.2 Clustering by semantic distance

Given a linear ordering, thumbnail-clustering exponentially increases going outwards from the focus. If no specific semantic distance among images is specified, thumbnails are considered equidistant to each other, and then clustered reflecting this exponential growth. However, to cluster together *similar* images, reflecting some similarity concept, is useful.

The stochastic approach proposed in Brivio et al. [47] ensures that more different images will have a greater probability of lying in separate clusters, while more similar ones will probably belong to more crowded clusters.

Semantic distance can be computed in any of the domains D_i already cited in the previous section. Often, it is useful to measure semantic distances in a different domain than D_O. We call the semantic domain D_S. In Photocloud 3D navigation context, we experimented that effective results are given with D_O being the calibration domain, and D_S being the color-spatial-layout domain.

As for ordering, Photocloud is not aware of which and how a semantic is computed. The system just reads-in the image semantic distance for each couple of consecutive (in the ordering) images and directly uses it to dynamically cluster images. The default semantic distance used in the current implementation is described in Section 6.8.1.

6.7 Memory management

Image thumbnails, high resolution images, and the 3D model compete for both RAM and GPU memory. Though small, even a few hundreds of 256^2 RGB pixel images require more memory than the one realistically available. Compression techniques could be adopted to reduce memory requirements, but at certain stages the problem would rise again. As images are meant to reside on a remote server, it is also preferable to introduce policies to select only the most meaningful subset of the images to avoid loading of unnecessary data.

We address the problem using the same priority based cache system described for the 3D model. We arbitrarily assign half of the resources to the 3D model, while the rest is shared by the images. The currently selected image and the preview thumbnail are always assigned the highest priority, while the other thumbnails are assigned a priority according to how large is their visible area. Images are stored (either remotely, or on local hard drive) in JPEG format to limit the required band broadness. When loading into RAM, images are converted into RAW format, for subsequent load into GPU. We experimented DXT1 conversion from JPEG into RAM memory to save both memory and GPU band. However, even exploiting efficient GPU compression algorithms, it is a rather time consuming operation, which downgrades the overall PhotoCloud performance.

6.8 Preprocessing of 2D and 3D datasets

Given the input datasets composed of a 3D model and a calibrated image collection, PhotoCloud requires to pre-processed those datasets. Specifically, the efficient rendering of the 3D model is based on the Nexus encoding, while image processing needs to be computed offline for algorithm complexity reasons.

Thanks to camera calibration, we associate to each image its average depth, computed on the depth buffer of the 3D model rendered from the image viewpoint (this average depth is used at rendering time). In another thread, image ordering and distances are computed. Each image is associated a descriptor, representing it in an abstract domain. These are then used to estimate a good ordering and the semantic distance among images.

6.8.1 Image descriptors and image distances

An image can be described in various ways. Typically, the time of shot is an intuitive descriptor for it, but other meaningful alternatives exist, for instance based on image processing, or exploiting the calibration information to contextualize the image in a multi-dimensional environment. Whichever the case, this information is extracted from images and locally stored before linear ordering the images in the descriptor domain and computing their semantic distance.

Currently, the PhotoCloud preprocessing implements the following image descriptors: the time-of-shot, extracted from the file EXIF; the color distribution, as a 16 entry histogram of colors; the spatial-color-layout, as the 4×4 down-sampled image; image position and orientation, as given by the calibration. Note that all color computations are made in LAB color-space.

Then, for each couple of images I_a and I_b, and for each descriptor D, a distance value $d_D(I_a, I_b)$ is computed and stored in a table. Euclidean distance is computed for each multi-dimensional data, but inside the calibration space. The position and orientation Euclidean distances are computed separately and then averaged together. Descriptors and distance table are stored in memory for later access by the ordering and semantic distance algorithms.

6.8.2 Ordering images and setting the semantic

As specified in Sec. 6.6.1, to order images according to a selected descriptor domain can be useful. Finding an Hamiltonian path through all images represented into that domain has impractical computational costs. Instead, we adopt the following heuristic:

1. Compute the path that, starting from a random image, connects every image I_g to its nearest I_n (minimum $d_D(I_g, I_n)$);

2. Cut the sequence along the longest arch;

3. Iteratively perform a few optimization steps, swapping a short subsequence of consecutive images whenever this is found to shorten the path, until no such moves are detected.

Step 1 clearly introduces an approximation. To obtain better results, we perform the last two steps, whose goal is to optimize the initial raw ordering without downgrading the quadratic trend of the algorithm.

Once ordered, each image can be assigned a semantic distance from its previous image. The semantic domain can be chosen independently from the ordering one, but should be shared by all the images. Starting from the first image and following the order, the consecutive images distance is read from the descriptor distance table.

6.8.3 Index file format

PhotoCloud finally reads-in an XML-like index file, which specifies the list of images and the filename of the 3D model together with the attributes needed for the browsing. Each image is described with a separate tag which contains the full resolution and the thumbnail (i.e. $max(w, h) = 256pixels$) filenames. Image cardinality and semantic distance from the previous image are stored as an integer and a floating value, respectively. A child tag includes all the intrinsic and extrinsic calibration parameters as well as the average pixels depth. Optional tags allow to specify additional features and settings, like the background color (uniform or shaded), and the initial view.

6.9 Implementation and evaluation

PhotoCloud has been implemented as an opensource project and the current prototype additionally achieves:

- support for the most common calibration and 2D/3D data formats;
- fulfillment of entry level HW resources, which allows for efficient executions on common personal computers;
- cross-platform source code (for Windows, Mac OS X, and Unix) is available for both PhotoCloud [344] and the multiresolution 3D model encoding [343].

We tested PhotoCloud on different machines, using five datasets:

- **Dubrovnik City** (see Figure 6.8): 6,844 photographs at different high resolutions and a cloud of 2 million points.
- **Bouvignes Castle** (see Figure 6.3): 97 photographs at a resolution of 2,000 × 2,000 pixels and a cloud of 350K points.
- **Michelangelo's David** (see Figure 6.4): 125 photographs at 2,336 × 3,504 resolution and a mesh of 56 million triangles.
- **Cavalieri Square** (see Figure 6.7): 458 photographs at 3,872 × 2,592 resolution and a mesh of 15 million triangles. (see Figure 6.7)
- **Signoria Square** (see Figure 5.9): 507 photographs at 2,592 × 1,728 resolution and a mesh of 65 million triangles

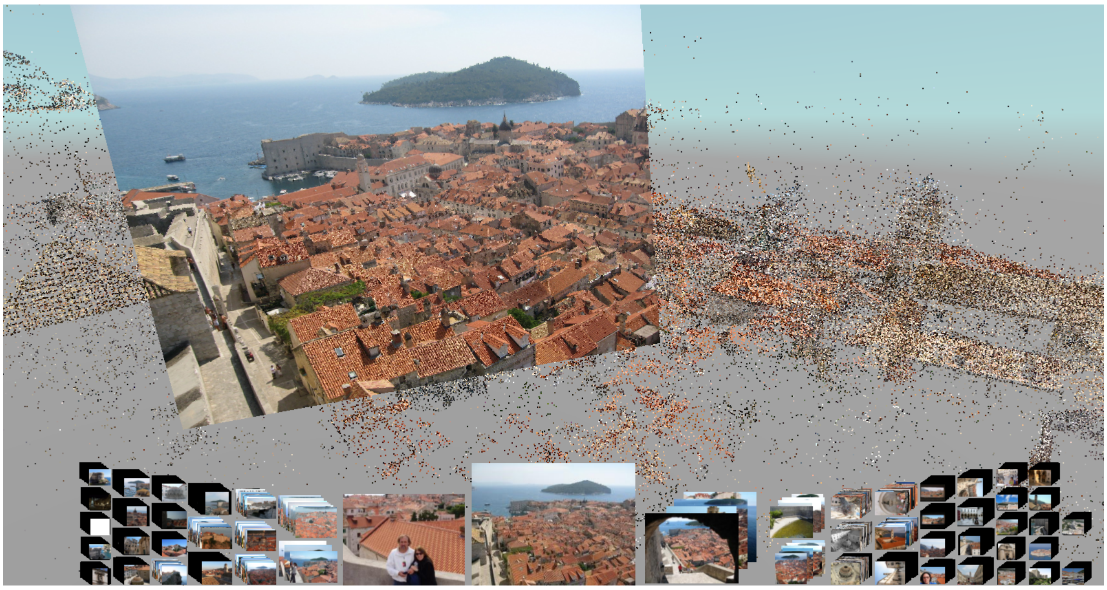


Figure 6.8: Screenshot of the Dubrovnik city.

In all cases, the client constantly achieved more than 60 frames per second. It occupied only approximately 128 Mbytes of RAM and 80 Mbytes of GPU memory on a laptop with 1,600 × 900 resolution, a 2.6-GHz dual-core processor, and an Nvidia GeForce GT 130M graphics card.

Whereas data loading is subject to network latency, the caching mechanism and incremental data structures optimize performance with respect to the underlying network layer's limitations. They require approximately 4 percent of CPU usage to handle data across the memory levels. In our tests, a standard 100-Mbit Ethernet network connection (with a peak nominal bandwidth of roughly 11.8 Mbytes per second) always provided the necessary bandwidth to keep latencies small.

We presented PhotoCloud to several Cultural Heritage experts, some of whom had no strong IT competence, and collected their impressions and comments. They all reported that the system was appealing and easy to use. Image-based navigation let the unskilled users avoid the "I'm lost" situation that often occurs when they face a 3D navigation system. In addition, the system's 3D navigation effectively helped the users select images.

6.10 User study

We quantitatively compared PhotoCloud's image-browsing interface with that of Photosynth [216], a publicly available Web-based implementation of Photo Tourism. We chose Photosynth because of its similar goals. User studies [225] on attitudes toward image browsing revealed that people tend to concentrate on events and thus on location cues. In our case, we wanted to evaluate the effectiveness of interaction mechanisms in a 3D environment.

6.10.1 The participants

Eighteen university students and young researchers participated. We separated them into three levels of self-assessed experience with 3D navigation: low, medium, and high. None had previously used either system, and none was familiar with the dataset. All had normal or corrected-to-normal vision with no color blindness.

6.10.2 The procedure

All the experiments took place under the same lighting conditions in a silent room. We allowed each participant a preliminary five-minute test run on each browser, using a training dataset. Each participant received a sheet with illustrated instructions about each tool's functionalities.

Then, each participant performed a sequence of tasks on the Cavalieri Square B dataset (a subset of the Cavalieri Square dataset). It featured a square with a statue in the middle and consisted of 202 photos and a point cloud recovered from the calibrated images.

A written assignment described the four tasks:

1. Read what is written on the front of the church (which required finding any of the five pictures featuring that writing).
2. Find any of the three images that feature the left staircase of a specific building.
3. Find any of the two pictures that feature that building's entire façade (i.e. both the left and right borders of the facade in a single image).
4. Determine whether there's an image showing the statue's back, and, if so, show it.

Timings started only after the participants read and understood each task. They were to work on each task until they completed it, and they received no assistance while performing the tasks.

The participants performed the tasks first on one system and then on the other. Although the two systems used the same dataset, the picture orders differed because our system computes the picture order as part of preprocessing. Because dataset knowledge clearly influences user performance, one randomly chosen half of the participants used PhotoCloud first; the other half used Photosynth first.

6.11 Results and discussion

Table 6.1 summarizes the results. Scene familiarity turned out to be not very important because the times did not change excessively according to which system

Table 6.1: The average time the participants took to complete each task. The second column indicates how familiar the participants were with 3D interfaces. The best results are in bold.

Task	Participant skill	Photosynth time [s] (A)	PhotoCloud time [s] (B)	B/A (%)
1	Low	**34.4**	96.0	279
	Medium	**29.0**	50.5	150
	High	**21.6**	31.1	130
2	Low	247.4	**29.2**	12
	Medium	240.5	**23.8**	10
	High	149.0	**13.0**	9
3	Low	**59.2**	128.0	216
	Medium	47.7	**45.7**	96
	High	29.0	**23.6**	81
4	Low	**58.0**	71.4	123
	Medium	45.7	**24.8**	54
	High	29.0	**15.6**	54

the participants used first. As we expected, the times improved with the participants' skill level, particularly with PhotoCloud. The participants' ability with the keyboard-and-mouse interface significantly affected their performance with PhotoCloud. However, this partly contrasts with our original aim, because we intended PhotoCloud for a broad class of users, including both computer science and cultural-heritage people.

Considering each task separately, the differences in the times are due partly to the different 3D-navigation mechanisms and partly to the visualization techniques. In Photosynth, the images' positions constrain the movement of the virtual 3D view, whereas PhotoCloud supports free view selection through keyboard-plus-mouse and framelet interactions. In general, PhotoCloud allows users a larger variety of actions. For example, they can view the 3D model from points and angles not pictured in any image. Or, they can virtually walk in the 3D environment toward the part of the scene they're interested in before selecting the target image.

Our intent is that this approach should reasonably reduce the time to complete the tasks. However, the participants' performance noticeably deteriorated when they solved tasks in which the target image had to match a specific view (tasks 1 and 3). This was mainly because we displayed the framelets at 0.1 × *depthC* to prevent cluttering in areas with a higher density of shots.

Specifically, while solving task 1 with PhotoCloud, the participants tended to move near the front of the church instead of selecting an image and zooming in to read the writing. This resulted in longer times. A similar misunderstanding occurred during task 3. In these cases, constraining the view to the available images can noticeably decrease the times (by nearly two-thirds, for task 1 for low-skilled users), due to the spatial metaphor. However, this happens only if the frustums of that image and other images intersect.

In contrast, free 3D movements can reduce the search time up to 90 percent. In task 2, each requested picture featured the staircase (occupying the largest part of the picture) in the foreground and a more distant building in the background. During the tests, all but one participant initially used the Photosynth 3D browser and overhead map to reach that picture but finally relied on the 2D image browser to find it. With PhotoCloud, moving near the desired location and selecting the correct framelet accomplished the task. During the experiments, we registered which interface mechanisms each user tried and which one was ultimately successful. In the 75% of the cases with PhotoCloud, framelets were successful, but their use always followed either 3D free navigation (90%) or 2D browsing (10%). In 20% of the experiments, the participants used mainly the embedded 2D browser. On the other hand, when using Photosynth, the participants often switched between the 3D view and the overhead map, sometimes resorting to the conventional 2D browser, which proved time-consuming. With PhotoCloud, the participants mostly used the 2D-3D interface; the integration of the various tools in the interface helped them switch between different, effective navigation strategies.

After the test, we asked the participants for qualitative comments and impressions; most argued that picture localization was more natural and easier in PhotoCloud. As they pointed out, this is probably due to the ability to freely move in the scene in PhotoCloud. Photosynth only lets users jump from one picture to another, which isn't always the one they expect. The participants also reported that PhotoCloud's visualization techniques helped them better understand how the scene was structured, which objects were in it, and how to reach them.

Chapter 7

Conclusion

The use of multi-view image data sets has gained importance in the computer vision and computer graphics research communities in the last years. This has lead to several new applications, and to the improvement of algorithms which were typically designed for a single image. This thesis follows the path of this research trend, presenting different scientific contributions related to the reconstruction, registration and visualization of multi-view image data sets.

A pre-processing algorithm has been proposed. This exploits the global color information of the image set in order to convert images for improving the result of image-based reconstruction algorithms; dense stereo matching in particular. Key results from this study are:

- unsharp masking is very important for the matching performance;
- the performance of the classic unsharp masking and the C-USM demonstrates that standard USM is powerful enough for matching purposes
- unsharp masking applied to CIE-Y can be the best compromise between ease of implementation and performance obtained.

Despite the last result, we want to underline that the proposed Multi-Image Decolorize (MID) algorithm is often the best conversion in terms of matching when unsharp masking is not applied. Moreover, we found the idea to provide reconstruction improvements through smarter pre-processing of the input image set is very interesting for two main reasons:

- the pre-processing is not related to a particular algorithm, but many reconstruction algorithms can benefit from this;
- we have showed that a proper use of color information is at the moment under-explored. This is due to the fact that a suitable color-to-gray conversion can have similar performance of the algorithms that use colors.

One of the most interesting future research directions, that this work suggests, is the study of a grayscale conversion for image matching that does not rely on the properties of the existing methods; but it follows different paradigm even at the cost of heavily-reduce the image quality of the converted image from a perceptual point of view. Furthermore, novel enhanced pre-processing operations are interesting to be investigated and developed. They do not have to be limited to an optimal aggregation/exploitation of the color information, but they need to exacerbate details to further increase the performance of stereo and multi-view reconstruction algorithms.

Another pre-processing step has been proposed, it exploits information related to an entire calibrated image set to provide a color mapping without shadows artifacts. Even if this is a smaller contribution with respect to the others, this technique is very useful from an application point of view, because color mapping of outdoor scene is often achieved through weighted blending scheme.

The classical problem of image-based localization has been recast here in a large scale 2D/3D registration problem in order to obtain both image localization and very high accuracy of the position and orientation of the localized shot. Moreover, the 2D/3D registration and localization algorithm is supported by a novel validation procedure. This makes the system able to validate in an unsupervised way the results obtained, and very robust against false positive registration. This strives for novel tourism applications able to contextualize and visualize the photographs during the visit of important monuments such as buildings, squares, etc. Potential improvements of the system are the refinement of the growth of the calibration support dataset by employing global registration methods such as the feature-based method by Stamos et al. [314] or the statical method by Corsini et al. [73], and studying different strategies in order to speedup the calibration stage.

Finally, a novel visualization system for the presentation and the navigation of calibrated image dataset, PhotoCloud, has been proposed. Despite it shares some similarity with PhotoSynth, PhotoCloud has unique characteristics such as: the independence from a specific 3D model type, a strong tolerance to both 2D/3D and 2D/2D inconsistencies in the data, and a set of effective 3D navigation controls. Furhtermore, it is tightly integrated with innovative thumbnail bars that employ clustering to present the maximum amount of image information in a compact screen space. Interesting research directions are the use of different clustering approaches and to exploit high level image content analysis. For example, automatic tagging can enhance the visualization/presentation of the taken photographs. Other improvements of the visualization systems may be the precomputation of image paths to produce better transitions between images, and collision detection system to help the 3D navigation of a scene. Another interesting development would be a timeline to navigate between historic photographs on their related monuments/place of interest.

7.1 Comments on contributions

The presented contributions follow the chronological order in which the author worked on them. The first two contributions results, however, are not used in the other system. This is because of the differences in the characteristics of the faced problems, as listed in the first three items of the list in Sec. 1.1:

The grayscale conversion algorithm has been tested specifically in the case of dense stereo matching (like Fig. 1.1), and it does not trivially transpose in more advanced multi-view stereo environments (like Fig. 1.2 or Fig. 1.3), where different measures, e.g. Normalized Cross Correlation, are preferably used to sparsely match image correspondences. Thus, a new experimental setup is needed to validate the advantage of a grayscale preprocessing.

The shadow removal algorithm is specific for laser-scanned 3D acquisition campaigns where the photographic campaign is performed professionally, e.g. Fig. 1.2, and both presence and correctness of the EXIF information are certain. This is extremely different from the case of the subsequent contributions, where the photographs come from non-controlled environments, e.g. Fig. 1.3.

On the other hand, the unsupervised image-based localization algorithm has been designed specifically to cooperate with the PhotoCloud visualization system in the same non-controlled environment, eg.Fig. 1.3.

The thesis author is first author of the publications relative to the grayscale conversion and the unsupervised image-based localization algorithms, and is second author in the publications relative to the other contributions. For the shadow removal algorithm, the author contribution lies mainly in the actual photographic corrections of images once a shadow mask has been computed. Concerning PhotoCloud, the first paper author did the main development and design efforts. This thesis author contributions have been fundamental and distributed along all the system design and implementation; as often happens in many large and complex systems. One of the significant portions of the system, that was implemented by this thesis author, is the management of the color in the rendering pipeline, described in Sec. 6.3.3; for this aspect various approaches have been implemented and evaluated in order to provide the most efficient solution.

7.2 List of publications

The research contribution presented in this thesis was the subject of the following publications:

- Benedetti Luca, Corsini Massimiliano, Cignoni Paolo, Callieri Marco, and Scopigno Roberto.
  Color to gray conversions in the context of stereo matching algorithms. An analysis and comparison of current methods and an ad-hoc theoretically-motivated technique for image matching.
  In Machine Vision and Applications, pages 1–22, Springer Berlin / Heidelberg.
  DOI: 10.1007/s00138-010-0304-x [22]

- Dellepiane Matteo, Benedetti Luca, and Scopigno Roberto.
  Removing shadows for color projection using sun position estimation.
  In 11th VAST International Symposium on Virtual Reality, Archaeology and Cultural Heritage, page 55–62, Eurographics.
  DOI: 10.2312/VAST/VAST10/055-062 [87]

- Benedetti Luca, Corsini Massimiliano, Dellepiane Matteo, Cignoni Paolo, and Scopigno Roberto.
  GAIL: Geometry-aware Automatic Image Localization.
  In VISAPP 2013 - International Conference on Computer Vision Theory and Applications, Number in press - 2013
  DOI: 10.5220/0004281800310040 [23]

- Brivio Paolo, Benedetti Luca, Tarini Marco, Ponchio Federico, Cignoni Paolo, and Scopigno Roberto.
  PhotoCloud: Interactive Remote Exploration of Joint 2D and 3D Datasets.
  In IEEE Computer Graphics and Applications, vol. 33, no. 2, pp. 86-96, c3, March-April 2013
  DOI: 10.1109/MCG.2012.92 [45]

Bibliography

[1] E. H. Adelson, E. P. Simoncelli, and R. Hingorani. Orthogonal pyramid transforms for image coding. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 845, pages 50–58, 1987. 14

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *Proceedings of International Conference on Computer Vision*, 2009. 18, 21

[3] T. Akenine-Möller, E. Haines, and N. Hoffman. *Real-Time Rendering 3rd Edition.* A. K. Peters, Ltd., Natick, MA, USA, 2008. 25

[4] A. Alsam and Ø. Kolås. Grey colour sharpening. In *Fourteenth Color Imaging Conference*, pages 263–267, Scottsdale, Arizona, Nov 2006. 36, 37

[5] G. Amato and F. Falchi. kNN based image classification relying on local feature similarity. In *Proc. SISAP'10*, pages 101–108. ACM, 2010. 80

[6] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, 1989. 14

[7] E. Arbel and H. Hel-Or. Texture-preserving shadow removal in color images containing curved surfaces. In *Conference on Computer Vision and Pattern Recognition (CVPR 2007).* IEEE Computer Society, June 2007. 73

[8] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. 16

[9] K. B. Atkinson. *Close range photogrammetry and machine vision.* Whittles, Scotland, UK, 1996. 17

[10] H. Aydinoglu and M. H. Hayes III. Compression of multi-view images. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 2, pages 385–389. IEEE, 1994. 5

[11] M. A. Badamchizadeh and A. Aghagolzadeh. Comparative study of unsharp masking methods for image enhancement. *International Conference on Image and Graphics*, 0:27–30, 2004. 50

[12] H. H. Baker and T. O. Binford. Depth from edge and intensity based stereo. In *IJCAI81*, pages 631–636, 1981. 15

[13] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 14

[14] R. Bala and R. Eschbach. Spatial color-to-grayscale transform preserving chrominance edge information. In *Color Imaging Conference*, pages 82–86, 2004. 33, 36, 37

[15] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice Hall, Englewood Cliffs, New Jersey, 1982. 14

[16] N. Bannai, A. Agathos, and R. B. Fisher. Fusing multiple color images for texturing models. In *3DPVT04*, pages 558–565, 2004. 22, 23

[17] S. T. Barnard and M. A. Fischler. Computational stereo. *ACM Comput. Surv.*, 14, December 1982. 15

[18] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994. 14

[19] A. Baumberg. Blending images for texturing 3D models. In *Proceedings of the British Machine Vision Conference 2002, BMVC 2002*, pages 404–413. British Machine Vision Association, 2002. 23

[20] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Proceedings*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer, 2006. 16

[21] B. B. Bederson. Photomesa: a zoomable image browser using quantum treemaps and bubblemaps. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 71–80, New York, NY, USA, 2001. ACM. 26

[22] L. Benedetti, M. Corsini, P. Cignoni, M. Callieri, and R. Scopigno. Color to gray conversions in the context of stereo matching algorithms. *Machine Vision and Applications*, 23(2):327–348, 2012. 7, 120

[23] L. Benedetti, M. Corsini, M. Dellepiane, P. Cignoni, and R. Scopigno. Gail: Geometry-aware automatic image localization. In *VISAPP 2013 - International Conference on Computer Vision Theory and Applications (in press)*, page 10. Springer, 2013. 10, 120

[24] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Computer VisionÄîECCV'92*, pages 237–252. Springer, 1992. 14

[25] F. Bernardini, I. M. Martin, and H. Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics*, 7(4):318–332, October 2001. 23

[26] R. S. Berns. *Billmeyer and Saltzman's Principles of Color Technology.* Wiley - Interscience, third edition, 2000. 33

[27] M. Bertero, T. A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, 1988. 14

[28] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293, 1999. 54

[29] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. In *Motion and Video Computing, 2002. Proceedings. Workshop on*, pages 169–174. IEEE, 2002. 5

[30] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996. 14

[31] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91, 1996. 54

[32] A. Blake, P. Kohli, and C. Rother. *Advances in Markov Random Fields for Vision and Image Processing.* MIT Press, 2010. 14

[33] A. Blake, A. Zimmerman, and G. Knowles. Surface descriptions from stereo and shading. *Image Vision Comput.*, 3:183–191, November 1986. 14

[34] A. Blake and A. Zisserman. Visual reconstruction. *Mit Press Series In Artificial Intelligence*, page 225, 1987. 14

[35] M. Bleyer, S. Chambon, U. Poppe, and M. Gelautz. Evaluation of different methods for using colour information in global stereo matching approaches. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXVII, part B3a, pages 415–422, 2008. 39, 40

[36] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, pages 7–55, 1987. 14

[37] R. Bouwmeester. Sunposition.com.
`http://www.sunposition.com/`. 71

[38] K. W. Bowyer, C. Kranenburg, and S. Dougherty. Edge detector evaluation using empirical roc curves. *Comput. Vis. Image Underst.*, 84:77–103, October 2001. 16

[39] I. Boyadzhiev, S. Paris, and K. Bala. User-assisted image compositing for photographic lighting. In *to be presented at SIGGRAPH 2013*, volume 32, 2013. 5

[40] E. Boyer and M. O.O. Berger. 3d surface reconstruction using occluding contours. *International Journal of Computer Vision*, 22(3):219–233, 1997. 19

[41] Y. Boykov, O. Veksler, and R. Zabih. A variable window approach to early vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1283–1294, 1998. 15

[42] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001. 14, 19

[43] M. Brackenridge. Sunposition calculator.
`http://sunposition.info/sunposition/index.php`. 71

[44] D. Bradley, T. Boubekeur, and W. Heidrich. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 20

[45] P. Brivio, L. Benedetti, M. Tarini, F. Ponchio, P. Cignoni, and R. Scopigno. Photocloud: Interactive remote exploration of joint 2d and 3d datasets. *IEEE Computer Graphics and Applications*, 33(2):86–96,c3, 2013. 11, 78, 120

[46] P. Brivio, M. Tarini, and P. Cignoni. Browsing large image datasets through voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization 2010)*, 16(6):1261–1270, November - December 2010. 26

[47] P. Brivio, M. Tarini, F. Ponchio, and P. Cignoni. Pilebars: Scalable dynamic thumbnail bars. Technical Report 2011-TR-006, ISTI-CNR, Pisa, Italy, 2011. 107, 109

[48] M. Brown and D. G. Lowe. Unsupervised 3d object recognition and reconstruction in unordered datasets. In *3D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*, pages 56–63, june 2005. 17, 18

[49] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61:211–231, 2005. 14

[50] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 425–432, New York, NY, USA, 2001. ACM. 26

[51] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on communications*, 31(4):532–540, 1983. 14

[52] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)*, 2(4):217–236, 1983. 14

[53] M. Čadík. Perceptual evaluation of color-to-grayscale image conversions. *Comput. Graph. Forum*, 27(7):1745–1754, 2008. 31, 52, 67

[54] M. Callieri, P. Cignoni, M. Corsini, and R. Scopigno. Masked photo blending: mapping dense photographic dataset on high-resolution 3d models. *Computer & Graphics*, 32(4):464–473, Aug 2008. 23, 70, 75

[55] M. Callieri, P. Cignoni, M. Corsini, and R. Scopigno. Technical section: Masked photo blending: Mapping dense photographic data set on high-resolution sampled 3d models. *Comput. Graph.*, 32:464–473, August 2008. 102

[56] M. Callieri, P. Cignoni, and R. Scopigno. Reconstructing textured meshes from multiple range rgb maps. In *7th Int.l Fall Workshop on Vision, Modeling, and Visualization 2002*, pages 419–426, Erlangen (D), Nov. 20 - 22 2002. IOS Press. 22

[57] N. D. F. Campbell and G. Vogiatzis. Automatic Object Segmentation from Calibrated Images. *2011 Conference for*, 2011. 19

[58] N. D. F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 766–779, Berlin, Heidelberg, 2008. Springer-Verlag. 21

[59] N. D. F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Automatic 3D object segmentation in multiple views using volumetric graph-cuts. *Image and Vision Computing*, 28(1):14–25, January 2010. 19

[60] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell*, 8(6):679–698, 1986. 16

[61] S. Chambon and A. Crouzil. Color stereo matching using correlation measures. In *Complex Systems Intelligence and Modern Technological Applications - CSIMTA 2004, Cherbourg, France*, pages 520–525. LUSAC, sep 2004. 39

[62] S. E. Chen and L. Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 279–288. ACM, 1993. 25

[63] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, pages 1–8, 2007. 4

[64] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. Meshlab: an open-source mesh processing tool. In *Sixth Eurographics Italian Chapter Conference*, pages 129–136, 2008. 80

[65] P. Cignoni, F. Ganovelli, E. Gobbetti, F. Marton, F. Ponchio, and R. Scopigno. Adaptive tetrapuzzles: efficient out-of-core construction and visualization of gigantic multiresolution polygonal models. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 796–803, New York, NY, USA, 2004. ACM. 101

[66] P. Cignoni, F. Ganovelli, E. Gobbetti, F. Marton, F. Ponchio, and R. Scopigno. Batched multi triangulation. In *Proceedings IEEE Visualization*, pages 207–214, Conference held in Minneapolis, MI, USA, October 2005. IEEE Computer Society Press. 107

[67] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision*, 9(2):83–112, 1992. 19

[68] R. Cipolla and P. J. Giblin. *Visual motion of curves and surfaces*. Cambridge University Press, 2000. 19

[69] R. Cipolla, D. Robertson, and B. Tordoff. Image-based localisation. In *Proc. of 10th Int. Conf. on Virtual Systems and Multimedia*, pages 22–29, 2004. 24

[70] M. B. Clowes. On seeing things. *Artificial Intelligence*, 2(1):79–116, 1971. 14

[71] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. 3d urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision*, 78(2):121–141, 2008. 17

[72] M. Corsini, M. Callieri, and P. Cignoni. Stereo light probe. *Computer Graphics Forum*, 27(2):291–300, 2008. 71

[73] M. Corsini, M. Dellepiane, F. Ganovelli, R. Gherardi, A. Fusiello, and R. Scopigno. Fully automatic registration of image sets on approximate geometry. *Int. J. Comput. Vision*, 102(1-3):91–111, 2013. 118

[74] M. Corsini, M. Dellepiane, F. Ponchio, and R. Scopigno. Image-to-geometry registration: a mutual information method exploiting illumination-related geometric properties. *Computer Graphics Forum*, 28(7):1755–1764, October 2009. 72

[75] R. Crane. *Simplified Approach to Image Processing: Classical and Modern Techniques in C.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 1996. 16

[76] Y. Cui, N. Hasler, T. Thormählen, and H.-P. Seidel. Scale invariant feature transform with irregular orientation histogram binning. In *Image Analysis and Recognition, 6th International Conference, ICIAR 2009. Proceedings*, volume 5627, pages 258–267. Springer, 2009. 16

[77] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA, 1996. ACM. 20

[78] P. Daras and A. Axenopoulos. A compact multi-view descriptor for 3d object retrieval. In *Content-Based Multimedia Indexing, 2009. CBMI'09. Seventh International Workshop on*, pages 115–119. IEEE, 2009. 4

[79] A. Davis, M. Levoy, and F. Durand. Unstructured light fields. In *Computer Graphics Forum*, volume 31, pages 305–314. Wiley Online Library, 2012. 26

[80] L. S. Davis. A survey of edge detection techniques. *Computer Graphics and Image Processing*, 4(3):248–270, 1975. 14, 16

[81] J. S. De Bonet and P. Viola. Poxels: Probabilistic voxelized volume reconstruction. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 418–425, 1999. 19

[82] R. L. de Queiroz and K. M. Braun. Color to gray and back: color embedding into textured gray images. *IEEE Transactions on Image Processing*, 15(6):1464–1470, 2006. 33, 37, 39

[83] P. Debevec, Y. Yu, and G. Boshokov. Efficient view-dependent image-based rendering with projective texture-mapping. Technical report, University of California at Berkeley, Berkeley, CA, USA, 1998. 102

[84] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 11–20, New York, NY, USA, 1996. ACM. 26

[85] M. Deering, S. Winner, Bic S., C. Duffy, and N. Hunt. The triangle processor and normal vector shader: a vlsi system for high performance graphics. *SIGGRAPH Comput. Graph.*, 22(4):21–30, 1988. 102

[86] A. Delaunoy, E. Prados, P. G. I. Piracés, J.-P. Pons, P. Sturm, et al. Minimizing the multi-view stereo reprojection error for triangular surface meshes. In *British machine vision conference*, 2008. 20

[87] M. Dellepiane, L. Benedetti, and R. Scopigno. Removing shadows for color projection using sun position estimation. In *VAST10: The 11th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, volume Full Papers, pages 55–62, Paris, France, 09/2010 2010. Eurographics Association, Eurographics Association. 8, 120

[88] M. Dellepiane, M. Callieri, M. Corsini, P. Cignoni, and R. Scopigno. Improved color acquisition and mapping on 3d models via flash-based photography. *ACM Journal on Computing and Cultural Heritage*, 2(4):article n. 9 20, 2010. In: ACM Journal on Computing and Cultural Heritage, vol. 2 (4) article n. 9. ACM, 2010. 23, 71, 72

[89] R. Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1:167–187, 1987. 16

[90] P. Dev. Perception of depth surfaces in random-dot stereograms: a neural model. *International Journal of Man-Machine Studies*, 7(4):511–528, 1975. 15

[91] E. D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1(4):223–240, 1988. 14

[92] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. In *ACM Trans. on Graphics (Proceedings of Siggraph Conference)*, volume 23. ACM Press, 2004. 23

[93] P. Eisert, E. Steinbach, and B. Girod. Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views. *Circuits and Systems for Video Technology, IEEE Transactions on*, 10(2):261–277, 2000. 19

[94] B. Epshtein, E. Ofek, Y. Wexler, and P. Zhang. Hierarchical photo organization using geo-relevance. In *GIS '07: Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, pages 1–7, New York, NY, USA, 2007. ACM. 26

[95] M. D. Fairchild. *Color Appearance Models.* Addison-Wesley, second edition, 2005. 33, 34, 38

[96] M. D. Fairchild and E. Pirrotta. Predicting the lightness of chromatic object colors using cielab. *Color Research & Application*, 16(6):385–393, 1991. 34

[97] M. Farenzena, A. Fusiello, R. Gherardi, and R. Toldo. Towards unsupervised reconstruction of architectural models. In *VMV*, pages 41–50, 2008. 18

[98] O. D. Faugeras. *Three-dimensional computer vision: a geometric viewpoint.* MIT Press, Cambridge, MA, USA, 1993. 14

[99] O. D. Faugeras and R. Keriven. Variational principles, surface evolution, PDEs, level set methods, and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, 1998. 19

[100] O. D. Faugeras, Q. T. Luong, and T. Papadopoulo. *The geometry of multiple images*, volume 2. MIT press Cambridge, 2001. 17

[101] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation from single or multiple model views. *International Journal of Computer Vision*, 67(2):159–188, 2006. 4

[102] G. D. Finlayson, M. S. Drew, and Cheng Lu. Intrinsic images by entropy minimization. In *In Lecture Notes in Computer Science: Proc. 8th European Conference on Computer Vision (ECCV)*, volume 3023, pages 582–595, Praque, 2004. Springer. 23, 73

[103] G. D. Finlayson, M. S. Drew, and Cheng Lu. Entropy minimization for shadow removal. *Int. J. C. Vision*, 85(1):35–57, 2009. 73

[104] M. A. Fischler and O. Firschein. *Readings in computer vision: issues, problems, principles, and paradigms.* Morgan Kaufmann, 1987. 14

[105] M. A. Fischler, P. V. Fua, S. T. Barnard, O. Firschein, and Y. G. Leclerc. The vision problem: Exploiting parallel computation, 1989. 14

[106] 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):381–395, June 1981. 82

[107] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *European Conference on Computer Vision*, pages 311–326. Springer-Verlag, 1998. 17, 18

[108] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002. 14

[109] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 368–381, Berlin, Heidelberg, 2010. Springer-Verlag. 21

[110] T. Franken, M. Dellepiane, F. Ganovelli, P. Cignoni, C. Montani, and R. Scopigno. Minimizing user intervention in registering 2d images to 3d models. *The Visual Computer*, 21(8-10):619–628, sep 2005. Special Issues for Pacific Graphics 2005. 72

[111] C. Fredembach and G. D. Finlayson. Simple shadow removal. In *18th International Conference on Pattern Recognition (ICPR)*, volume 1, pages 832–835, Hong Kong, China, August 2006. IEEE Computer Society. 23, 73

[112] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:891–906, September 1991. 16

[113] N. Friedman and S. Russell. Image segmentation in video sequences: a probabilistic approach. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, UAI'97, pages 175–181, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. 20

[114] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction: combining multi-image stereo and shading. *Int. J. Comput. Vision*, 16:35–55, September 1995. 19

[115] T. Fujii and H. Harashima. 3-d image coding based on affine transform. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, pages V–577. IEEE, 1994. 5

[116] Y. Furukawa. *High-fidelity image-based modeling*. ProQuest, 2008. 21

[117] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1434–1441, 2010. 21

[118] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 21

[119] Y. Furukawa and J. Ponce. Accurate camera calibration from multi-view stereo and bundle adjustment. *International Journal of Computer Vision*, 84(3):257–268, 2009. 21

[120] Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. *Int. J. Comput. Vision*, 81:53–67, January 2009. 19

[121] Y. Furukawa and J. Ponce. Patch-based multi-view stereo software. `http://www.di.ens.fr/pmvs/`, 2010. 21

[122] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, and D. Cohen-Or. Seamless montage for texturing models. *Computer Graphics Forum*, 29(2):479–486, 2010. 23

[123] P. Gargallo, E. Prados, and P. Sturm. Minimizing the reprojection error in surface reconstruction from images. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 20

[124] N. Gehrig and P. L. Dragotti. Distributed compression of multi-view images using a geometrical coding approach. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 6, pages VI–421. IEEE, 2007. 5

[125] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984. 14

[126] S. Gibson, J. Cook, T. Howard, R. Hubbold, and D. Oram. Accurate camera calibration for off-line, video-based augmented reality. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, pages 37–, Washington, DC, USA, 2002. IEEE Computer Society. 18

[127] E. Gobbetti and F. Marton. Layered point clouds. In Marc Alexa, Markus Gross, Hanspeter Pfister, and Szymon Rusinkiewicz, editors, *Eurographics Symposium on Point Based Graphics*, pages 113–120, 227, Aire-la-Ville, Switzerland, June 2004. Eurographics Association. Conference held in Zurich, Switzerland, June 2–5, 2004. 107

[128] M. Goesele, J. Ackermann, S. Fuhrmann, C. Haubold, R. Klowsky, D. Steedly, and R. Szeliski. Ambient point clouds for view interpolation. In *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pages 95:1–95:6, New York, NY, USA, 2010. ACM. xii, 27, 28

[129] M. Goesele, B. Curless, and S. M. Seitz. Multi-view stereo revisited. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2402–2409. IEEE, 2006. 20

[130] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. *Computer Vision, IEEE International Conference on*, 0:1–8, 2007. 21

[131] C. Gonzalez, Rafael and E. Woods, Richard. *Digital Image Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, third edition, 2006. 37

[132] A. A. Gooch, S. C. Olsen, J. Tumblin, and B. Gooch. Color2gray: salience-preserving color removal. *ACM Trans. Graph.*, 24(3):634–639, July 2005. 33, 37, 38

[133] Google. Picasa.
`http://picasa.google.com/`, 2004. 26

[134] M. Gopi, S. Krishnan, and C.T. Silva. Surface reconstruction based on lower dimensional localized delaunay triangulation. *Computer Graphics Forum*, 19(3):467–478, 2000. 20

[135] I. Gordon and D. G. Lowe. What and where: 3d object recognition with accurate pose. *Toward category-level object recognition*, pages 67–82, 2006. 24

[136] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1996. 26, 73

[137] W. E. L. Grimson. Computational experiments with a feature based stereo algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-7(1):17–34, jan. 1985. 15

[138] M. Grundland and N. A. Dodgson. The decolorize algorithm for contrast enhancing, color to grayscale conversion. Technical Report UCAM-CL-TR-649, University of Cambridge, Computer Laboratory, October 2005. 29, 33, 37, 38

[139] M. Grundland and N. A. Dodgson. Decolorize: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recogn.*, 40(11):2891–2896, 2007. 33, 37, 38

[140] L. Gu, S. Z. Li, and H.-J. Zhang. Learning probabilistic distribution model for multi-view face detection. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–116. IEEE, 2001. 4

[141] Y. Guan, C.-T. Li, and Y. Hu. An adaptive system for gait recognition in multi-view environments. In *Proceedings of the on Multimedia and security*, pages 139–144. ACM, 2012. 5

[142] J. Guild. The colorimetric properties of the spectrum. *Philosophical Transactions of the Royal Society of London*, A230:149–187, 1931. 34

[143] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Optimizing color consistency in photo collections. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2013)*, 32(4):85:1 – 85:9, 2013. 5

[144] M. J. Hannah. *Computer matching of areas in stereo images.* PhD thesis, Stanford University, Stanford, CA, USA, 1974. 15

[145] M. J. Hannah. Test results from sri's stereo system. In *Science Applications International Corp, Proceedings: Image Understanding Workshop,*, volume 2, 1988. 15

[146] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15. Manchester, UK, 1988. 15, 24

[147] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004. 17, 30

[148] G. E. Healey, S. A. Shafer, and L. B. Wolff. *Color. Physics-based vision: principles and practice.* Jones and Bartlett Publishers, Inc., USA, 1992. 14

[149] M. D. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer. Comparison of edge detectors: a methodology and initial study. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 143–148. IEEE, 1996. 16

[150] C. Hernández Esteban and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004. 19

[151] V. H. Hiep, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1430–1437. IEEE, 2009. 19

[152] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007. 55

[153] M. Hofmann and D. M. Gavrila. Multi-view 3d human pose estimation combining single-frame recovery, temporal integration and model adaptation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2214–2221. IEEE, 2009. 5

[154] H. Hoppe and K. Toyama. Continuous flash. Technical Report MSR-TR-2003-63, Microsoft Research, 2003. 23

[155] B. K. P. Horn. Obtaining shape from shading information. In *PsychCV75*, pages 115–155, 1975. 14

[156] B. K. P. Horn. *Robot Vision*. McGraw-Hill Higher Education, 1st edition, 1986. 14

[157] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 1987. 83

[158] B. K. P. Horn and M. J. Brooks. The variational approach to shape from shading. *Computer Vision, Graphics, and Image Processing*, 33(2):174–208, 1986. 14

[159] B. K. P. Horn and M. J. Brooks. *Shape from Shading*. MIT Press, 1989. 14

[160] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981. 14

[161] B. K. P. Horn and E. J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, pages 51–76, 1988. 14

[162] M. Hu, Y. Wang, Z. Zhang, and D. Zhang. Multi-view multi-stance gait identification. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 541–544. IEEE, 2011. 5

[163] D. A. Huffman. Impossible objects as nonsense sentences. *Machine Intelligence*, 6:295–323, 1971. 14

[164] D. F. Huynh, S. M. Drucker, P. Baudisch, and C. Wong. Time quilt: scaling up zoomable photo browsers for large, unstructured photo collections. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1937–1940, New York, NY, USA, 2005. ACM. 26

[165] M.-H. Hyun, S.-Y. Kim, and Y.-S. Ho. Multi-view image matting and compositing using trimap sharing for natural 3-d scene generation. In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2008*, pages 397–400. IEEE, 2008. 5

[166] A. Iosifidis, A. Tefas, N. Nikolaidis, and I. Pitas. Multi-view human movement recognition based on fuzzy distances and linear discriminant analysis. *Computer Vision and Image Understanding*, 116(3):347–360, 2012. xi, 4, 5

[167] A. Irschara, C. Zach, and H. Bischof. Towards wiki-based dense city modeling. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 17, 18

[168] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, pages 2599–2606, 2009. 24, 25, 78

[169] M. Jancosek and T. Pajdla. Segmentation based multi-view stereo. In *Computer Vision Winter Workshop.(Cited on page 53.)*. Citeseer, 2009. 19

[170] G. Kamberov, G. Kamberova, O. Chum, Š. Obdržálek, D. Martinec, J. Kostkova, T. Pajdla, J. Matas, and R. Šára. 3D geometry from uncalibrated images. *Advances in Visual Computing*, pages 802–813, 2006. 17, 18

[171] T. Kanade. A theory of origami world. *Artificial Intelligence*, 13:279–311, June 1980. 14

[172] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16:920–932, September 1994. 15

[173] S. B. Kang, Y. Li, X. Tong, and H.-Y. Shum. Image-based rendering. *Found. Trends. Comput. Graph. Vis.*, 2(3):173–258, January 2006. 25

[174] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages 506–513, 2004. 16

[175] R. Kehl, M. Bray, and L. Van Gool. Full body tracking from multiple views using stochastic sampling. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 129–136. IEEE, 2005. 5

[176] P. Kohli and P. H. S. Torr. Dynamic graph cuts for efficient inference in markov random fields. *IEEE transactions on pattern analysis and machine intelligence*, 29(12):2079–2088, 2007. 14

[177] K. Kolev, T. Brox, and D. Cremers. Robust variational segmentation of 3D objects from multiple views. *Pattern Recognition*, pages 688–697, 2006. 19

[178] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1):80–96, 2009. 20

[179] K. Kolev, M. Klodt, T. Brox, D. Cremers, et al. Propagated photoconsistency and convexity in variational multiview 3d reconstruction. In *Proceedings of the First International Workshop on Photometric Analysis For Computer Vision-PACV 2007*, 2007. 20

[180] K. Kolev, T. Pock, and D. Cremers. Anisotropic minimal surfaces integrating photoconsistency and normal information for multiview stereo. In *European Conference on Computer Vision (ECCV)*, Heraklion, Greece, September 2010. 19

[181] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. In *ter Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, Ithaca, NY, USA, 2001. Cornell University. 56

[182] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, pages 82–96, London, UK, 2002. Springer-Verlag. 14, 19

[183] J. Kopf, B. Chen, R. Szeliski, and M. Cohen. Street slide: browsing street level imagery. *ACM Trans. Graph.*, 29:96:1–96:8, July 2010. xii, 27, 95

[184] K. Kraus and P. Waldhäusl. *Photogrammetry.* Dümmler, 1993. 17

[185] M. Kumar and P. H. S. Torr. Fast memory-efficient generalized belief propagation. *Ninth European Conference on Computer Vision (ECCV 2006)*, pages 451–463, 2006. 14

[186] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000. 19

[187] P.-Y. Laffont, A. Bousseau, S. Paris, F. Durand, G. Drettakis, et al. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics*, 31(6), 2012. 5

[188] J. F. Lalonde, A. A. Efros, and S. G. Narasimhan. Estimating natural illumination from a single outdoor image. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 183–190, sept. 2009. 71

[189] J. F. Lalonde, S. G. Narasimhan, and A. A. Efros. What do the sun and the sky tell us about the camera? *International Journal on Computer Vision*, 88(1):24–51, May 2010. 71

[190] P. Lambert and P. Hébert. Robust multi-view stereo without matching. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1614–1621. IEEE, 2009. 21

[191] M. J. Langford. *Advanced photography: a grammar of techniques.* Focal Press, Ltd., 1974. 50

[192] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 150–162, 1994. 19

[193] W. Lee, W. Woo, and E. Boyer. Identifying foreground from multiple images. In *Proceedings of the 8th Asian conference on Computer vision-Volume Part II*, pages 580–589. Springer-Verlag, 2007. 19

[194] J. Lengyel. The convergence of graphics and vision. *Computer*, 31(7):46–53, July 1998. 25

[195] H. Lensch, W. Heidrich, and H. Seidel. Automated texture registration and stitching for real world models. In *Proceedings of the 8th Pacific Graphics Conference on Computer Graphics and Application (PACIFIC GRAPHICS-00)*, pages 317–327. IEEE, October 3–5 2000. 22, 23

[196] S. Z. Li, Q. Fu, L. Gu, B. Scholkopf, Y. Cheng, and H. Zhang. Kernel machine based learning for multi-view face detection and pose estimation. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 674–679. IEEE, 2001. 4

[197] Y. Li, L. Gu, and T. Kanade. A robust shape model for multi-view car alignment. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2466–2473. IEEE, 2009. 5

[198] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, pages 791–804, 2010. xvii, 24, 25, 85

[199] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. *ACM Transactions on Graphics (TOG)*, 2:2–7, 2005. 20

[200] S. Lin, S. L. Yuanzhen, S. B. Kang, Xin Tong, and H. Y. Shum. Diffuse-specular separation and depth recovery from image sequences. In *In Proceedings of European Conference on Computer Vision (ECCV*, pages 210–224, 2003. 23

[201] T. Lindeberg. Scale-space for discrete signals. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:234–254, March 1990. 14

[202] A. Lippman. Movie-maps: An application of the optical videodisc to computer graphics. *SIGGRAPH Comput. Graph.*, 14:32–42, July 1980. 27

[203] L. Liu, J. Xing, and H. Ai. Multi-view vehicle detection and tracking in crossroads. In *Pattern Recognition (ACPR), 2011 First Asian Conference on*, pages 608–612. IEEE, 2011. 5

[204] W. Liu, W. Xu, L. Li, S. Li, H. Zhao, and J. Zhang. Improved mammographic mass retrieval performance using multi-view information. In *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*, pages 410–415. IEEE, 2010. 5

[205] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, pages 1150–1157, Washington, DC, USA, 1999. IEEE Computer Society. 15, 16

[206] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, November 2004. 80

[207] C. Lu, M. S. Drew, and G. D. Finlayson. Shadow removal via flash/noflash illumination. *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*, pages 198–201, Oct. 2006. 23

[208] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International joint conference on artificial intelligence*, volume 3, pages 674–679, 1981. 14

[209] M. Magnor. *Video-based Rendering.* A. K. Peters, 2005. 5

[210] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *Int. J. Comput. Vision*, 23:149–168, June 1997. 14

[211] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989. 14

[212] R. Mantiuk, K. Myszkowski, and H. P. Seidel. A perceptual framework for contrast processing of high dynamic range images. *ACM Transactions on Applied Perception*, 3(3):286–308, July 2006. 42

[213] D. Marr and T. A. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 1976. 15

[214] D. Marr and T. A. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 204(1156):301–328, 1979. 15

[215] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–238, 1989. 14, 54

[216] Microsoft. Photosynth.
`http://photosynth.net`, 2007. 28, 113

[217] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings of the Eighth International Conference On Computer Vision*, pages 525–531, 2001. 15, 16

[218] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005. 16

[219] D. Miyazaki, Y. Matsushita, and K. Ikeuchi. Interactive shadow removal from a single image using hierarchical graph cut. In Hongbin Zha, Rin-ichiro Taniguchi, and Stephen J. Maybank, editors, *Computer Vision - ACCV 2009*, Lecture Notes in Computer Science, pages 234–245. Springer, 2009. 73

[220] T. Moons, L. Van Gool, and M. Vergauwen. 3d reconstruction from multiple images part 1: Principles. *Found. Trends. Comput. Graph. Vis.*, 4:287–404, April 2010. 17

[221] H. P. Moravec. Towards automatic visual obstacle avoidance. In *IJCAI*, 1977. 15

[222] H. P. Moravec. The stanford cart and the cmu rover. *Proceedings of the IEEE*, 71(7):872–884, 1983. 15

[223] R. Morris and V. Smelyanskiy. Matching images to models - camera calibration for 3-d surface reconstruction. *Energy Minimization Methods*, pages 105–117, 2001. 24

[224] P. Musialski, P. Wonka, Daniel G. A., M. Wimmer, L. Van Gool, and W. Purgathofer. A survey of urban reconstruction. *EUROGRAPHICS State of the art reports*, 2012. 14

[225] M. Naaman, S. Harada, Q.-Y. Wang, H. Garcia-Molina, and A. Paepcke. Context data in geo-referenced digital photo collections. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 196–203, New York, NY, USA, 2004. ACM. 113

[226] M. Naaman, Y. J. Song, A. Paepcke, and H. Garcia-Molina. Automatic organization for digital photographs with geographic coordinates. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '04, pages 53–62, New York, NY, USA, 2004. ACM. 26

[227] H. H. Nagel. Image sequences: Ten (octal) years - from phenomenology towards a theoretical foundation. In *ICPR86*, pages 1174–1185, 1986. 14

[228] H. H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:565–593, September 1986. 14

[229] Y. Nakamura, T. Matsuura, K. Satoh, and Y. Ohta. Occlusion detectable stereo – occlusion patterns in camera matrix. In *CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pages 371–378, Washington, DC, USA, 1996. IEEE Computer Society. 55

[230] V. S. Nalwa. Edge-detector resolution improvement by image interpolation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9:446–451, May 1987. 16

[231] V. S. Nalwa. *A guided tour of computer vision.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1993. 14, 16

[232] V. S. Nalwa and T. O. Binford. On detecting edges. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:699–714, November 1986. 16

[233] P. J. Narayanan, P. W. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 3–10, Washington, DC, USA, 1998. IEEE Computer Society. 19

[234] S. K. Nayar, M. Watanabe, and M. Noguchi. Real-time focus range sensor. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18:1186–1198, December 1996. 14

[235] Y. Nayatani. Simple estimation methods for the Helmholtz-Kohlrausch effect. *Color Research & Application*, 22(6), 1997. 36

[236] Y. Nayatani. Relations between the two kinds of representation methods in the Helmholtz-Kohlrausch effect. *Color Research & Application*, 23(5), 1998. 36

[237] Y. Nayatani and H. Sakai. Confusion between observation and experiment in the Helmholtz-Kohlrausch effect. *Color Research & Application*, 33(3):250–253, 2008. 36

[238] L. Neumann, M. Čadík, and A. Nemcsics. An efficient perception-based adaptive color to gray transformation. In *Proceedings of Computational Aesthetics 2007*, pages 73–80, Banff, Canada, 2007. Eurographics Association. 33, 37, 38

[239] K. Ni, D. Steedly, and F. Dellaert. Out-of-core bundle adjustment for large-scale 3d reconstruction. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 18

[240] D. Nistér. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. *Computer Vision-ECCV 2000*, pages 649–663, 2000. 18

[241] R. D. Nowak and R. G. Baraniuk. Adaptive weighted highpass filters using multiscale analysis. *IEEE Transactions on Image Processing*, 7(7):1068–1074, 1998. 29, 38, 43

[242] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. Van Gool. Color-based object tracking in multi-camera environments. In *Pattern Recognition*, pages 591–599. Springer, 2003. 5

[243] University of Washington GRAIL Lab. Dubrovnik dataset.
`http://grail.cs.washington.edu/rome/dubrovnik/index.html`. 85

[244] Y. Ohta and T. Kanade. *Stereo by intra-and inter-scanline search using dynamic programming.* Carnegie-Mellon University, Dept. of Computer Science, 1983. 15

[245] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, 1993. 15

[246] F. Ortiz and F. Torres. Automatic detection and elimination of specular reflectance in color images by means of ms diagram and vector connected filters. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(5):681–687, Sept. 2006. 23

[247] L. Paletta, G. Fritz, C. Seifert, P.M. Luley, and A. Almer. A mobile vision service for multimedia tourist applications in urban environments. *2006 IEEE Intelligent Transportation Systems Conference*, pages 566–572, 2006. 24

[248] G. Palma, M. Callieri, M. Dellepiane, and R. Scopigno. A statistical method for svbrdf approximation from video sequences in general lighting conditions. *Computer Graphics Forum (Issue of Eurographics Symposium on Rendering 2012)*, 31(4):1491–1500, 2012. xi, 4

[249] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67:141–158, 2006. 14

[250] A. P. Pentland. Local shading analysis. *IEEE transactions on pattern analysis and machine intelligence*, 6(2):170–187, 1984. 14

[251] G. Petschnigg, R. Szeliski, M. Agrawala, M. F. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.*, 23(3):664–672, 2004. 23

[252] J. C. Platt, M. Czerwinski, and B. A. Field. Phototoc: Automatic clustering for browsing personal photographs. Technical Report MSR-TR-2002-17, Microsoft Research, 2002. 26

[253] T. A. Poggio, D. Geiger, T. Caw, W. Yang, J. Little, D. Weinshall, H. Biilthoff, A. Hurlbert, E. Gamble, M. Villalba, et al. The mit vision machine. In *Artificial Intelligence at MIT: Expanding Frontiers-Volume 2. MIT*, 1988. 14

[254] T. A. Poggio and C. Koch. Ill-posed problems in early vision: from computational theory to analogue networks. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 226(1244):303–323, 1985. 14

[255] T. A. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(6035):314–319, 1985. 14

[256] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. Pmf: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14(4):449–470, 1985. 15

[257] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999. 18

[258] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. N. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *Int. J. Comput. Vision*, 78:143–167, July 2008. 17

[259] F. Ponchio. *Multiresolution structures for interactive visualization of very large 3D datasets.* PhD thesis, Clausthal University of Technology, December 2008. 107

[260] J. P. Pons, R. Keriven, and O. D. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *Int. J. Comput. Vision*, 72:179–193, April 2007. 19, 20

[261] M. Potmesil. Generating octree models of 3d objects from their silhouettes in a sequence of images. *Computer Vision, Graphics, and Image Processing*, 40(1):1–29, 1987. 19

[262] K. Prazdny. Detection of binocular disparities. *Biological Cybernetics*, 52(2):93–99, 1985. 15

[263] K. Pulli, H. Abi-Rached, T. Duchamp, L. Shapiro, and W. Stuetzle. Acquisition and visualization of colored 3d objects. In *Proceedings of ICPR 98*, pages 11,15, 1998. 23

[264] K. Rajpoot, V. Grau, J. Alison Noble, H. Becher, and C. Szmigielski. The evaluation of single-view and multi-view fusion 3d echocardiography using image-driven segmentation and tracking. *Medical Image Analysis*, 15(4):514–528, 2011. 5

[265] V. Rankov, R. Locke, R. Edens, P. Barber, and B. Vojnovic. An algorithm for image stitching and blending. In *Proceedings of SPIE. Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XII*, volume 5701, pages 190–199, March 2005. 23

[266] K. Rasche, R. Geist, and J. Westall. Detail preserving reproduction of color images for monochromats and dichromats. *IEEE Comput. Graph. Appl.*, 25(3):22–30, 2005. 32, 33, 37, 38

[267] N. Razavi, J. Gall, and L. Van Gool. Backprojection revisited: Scalable multi-view object detection and similarity metrics for detections. In *Computer Vision–ECCV 2010*, pages 620–633. Springer, 2010. 5

[268] E. Reinhard, E. A. Khan, A. O. Akyz, and G. M. Johnson. *Color Imaging: Fundamentals and Applications.* A. K. Peters, Ltd., Natick, MA, USA, 2008. 33

[269] G. X. Ritter and J. N. Wilson. *Handbook of computer vision algorithms in image algebra.* CRC Press, 2nd edition, 2001. 16

[270] L. G. Roberts. *Machine Perception of Three-Dimensional Solids.* Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York, 1963. 14

[271] D. Robertson and R. Cipolla. An image-based system for urban navigation. In *Proc. BMVC*, volume 1, pages 260–272, 2004. 24

[272] A. Rosenfeld. Quadtrees and pyramids for pattern recognition and image processing. In *ICPR80*, pages 802–811, 1980. 14

[273] A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *SMC*, 6(6):420–433, June 1976. 14

[274] S. Rusinkiewicz and M. Levoy. Qsplat: a multiresolution point rendering system for large meshes. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 343–352, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 101

[275] D.-S. Ryu, W.-K. Chung, and H.-G. Cho. Photoland: a new image layout system using spatio-temporal information in digital photos. In *Proceedings of*

the 2010 ACM Symposium on Applied Computing, SAC '10, pages 1884–1891, New York, NY, USA, 2010. ACM. 26

[276] Y. Sato and K. Ikeuchi. Temporal-color space analysis of reflection. In *Computer Vision and Pattern Recognition*, pages 570–576, Jun 1993. 23

[277] J. T. Satre. Sunposition autoupdate. `http://satellite-calculations.com/Satellite/sunposauto.htm`. 71

[278] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *IEEE International Conference on Computer Vision (ICCV)*, pages 667–674, nov. 2011. xvii, 24, 25, 78, 85

[279] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *European Conference on Computer Vision*, volume 1, pages 414–431. Springer-Verlag, 2002. 15, 24

[280] D. Scharstein and C. Pal. Learning conditional random fields for stereo. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007. 55

[281] D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, 28(2):155–174, 1998. 54

[282] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, 2002. xi, 1, 29, 30, 31, 54, 56

[283] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2007)*, pages 1–7. IEEE Computer Society, 2007. 24

[284] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 489–498. ACM Press/Addison-Wesley Publishing Co., 2000. 5

[285] M. Segal, C. Korobkin, R. van Widenfelt, J. Foran, and P. Haeberli. Fast shadows and lighting effects using texture mapping. *SIGGRAPH Comput. Graph.*, 26:249–252, July 1992. 102

[286] S. M. Seitz, B. Curless, and J. Diebel. Middlebury multi-view stereo datasets,. `http://vision.middlebury.edu/mview/`, 2006. 19

[287] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 519–528. IEEE, 2006. 19

[288] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999. 19

[289] J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 231–242, New York, NY, USA, 1998. ACM. 25

[290] S. A. Shafer, G. E. Healey, and L. B. Wolff. *Physics-based vision: principles and practice.* Jones and Bartlett Publishers, Inc., USA, 1992. 14

[291] F. Shao, G.-Y. Jiang, M. Yu, and Y.-S. Ho. Fast color correction for multi-view video by modeling spatio-temporal variation. *Journal of Visual Communication and Image Representation*, 21(5):392–403, 2010. 5

[292] H. Shao, T. Svoboda, T. Tuytelaars, and L. Van Gool. HPAT indexing for fast object/scene recognition based on local appearance. *CIVR'03*, pages 307–312, 2003. 24

[293] G. Sharma. *Digital Color Imaging Handbook.* CRC Press, Inc., Boca Raton, FL, USA, 2002. 33

[294] H. L. Shen, H. G. Zhang, S. J. Shao, and J. H. Xin. Chromaticity-based separation of reflection components in a single image. *Pattern Recogn.*, 41(8):2461–2469, 2008. 23

[295] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. *Computer Science Tech. Report*, pages 94–125, August 1994. 42

[296] Y. Shor and D. Lischinski. The shadow meets the mask: Pyramid-based shadow removal. *Computer Graphics Forum*, 27(2):577–586, apr 2008. 73

[297] H.-Y. Shum, S.-C. Chan, and S. B. Kang. *Image-based rendering.* Springer Science+ Business Media, 2007. 5, 25

[298] H.-Y. Shum and L.-W. He. Rendering with concentric mosaics. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 299–306, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 26

[299] H.-Y. Shum, Q. Ke, and Z. Zhang. Efficient bundle adjustment with virtual key frames: A hierarchical approach to multi-frame structure from motion. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* IEEE, 1999. 18

[300] E. P. Simoncelli and E. H. Adelson. Non-separable extensions of quadrature mirror filters to multiple dimensions. *PIEEE*, 78(4):652–664, April 1990. 14

[301] E. P. Simoncelli and E. H. Adelson. Subband transforms. *SubCoding*, pages 143–192, 1990. 14

[302] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multi-scale transforms. *Information Theory, IEEE Transactions on*, 38(2):587–607, 1991. 14

[303] S. N. Sinha and M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*, pages 349–356, Washington, DC, USA, 2005. IEEE Computer Society. 19

[304] G. G. Slabaugh, W. B. Culbertson, T. Malzbender, M. R. Stevens, and R. W. Schafer. Methods for volumetric reconstruction of visual scenes. *Int. J. Comput. Vision*, 57:179–199, May 2004. 19

[305] C. C. Slama, C. Theurer, and S. W. Henriksen. *Manual of photogrammetry.* American Society of Photogrammetry Falls Church, Virginia, 1980. 17

[306] K. Smith, P. E. Landes, J. Thollot, and K. Myszkowski. Apparent greyscale: A simple and fast conversion to perceptually accurate images and video. *Computer Graphics Forum (Proceedings of Eurographics 2008)*, 27(2), apr 2008. 29, 33, 34, 36, 37, 38, 43, 52

[307] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4):56, December 1986. 24, 78

[308] N. Snavely, R. Garg, Steven M. Seitz, and R. Szeliski. Finding paths through the world's photos. *ACM Trans. Graph.*, 27:15:1–15:11, August 2008. 27, 28

[309] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25:835–846, July 2006. xii, 17, 18, 24, 27, 28, 78, 80, 81, 95

[310] N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *In Conference on Computer Vision and Pattern Recognition (CVPR*, pages 1–8, 2008. 18

[311] M. Solh and G. AlRegib. Miqm: A novel multi-view images quality measure. In *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*, pages 186–191. IEEE, 2009. 5

[312] M. Sormann, C. Zach, and K. Karner. Graph Cut Based Multiple View Segmentation for 3D Reconstruction. *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pages 1085–1092, June 2006. 19

[313] P. Srinivasan, P. Liang, and S. Hackwood. Computational geometric methods in volumetric intersection for 3d reconstruction. *Pattern Recognition*, 23(8):843–857, 1990. 19

[314] I. Stamos, L. Liu, C. Chen, G. Wolberg, G. Yu, and S. Zokai. Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes. *Int. J. Comput. Vision*, 78:237–260, July 2008. 118

[315] D. Steedly, I. Essa, and F. Dellaert. Spectral partitioning for structure from motion. In *Proc. Int. Conf. on Computer Vision*, pages 649–663, 2003. 18

[316] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 213–220. IEEE, 2009. 5

[317] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5(3):271–301, 1990. 14

[318] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 58(1):23–32, 1993. 19

[319] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010. 14, 17

[320] R. Szeliski and S. B. Kang. Recovering 3d shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, 1994. 17

[321] R. Szeliski and R. Weiss. Robust shape recovery from occluding contours using a linear smoother. *International Journal of Computer Vision*, 28(1):27–44, 1998. 19

[322] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1068–1080, 2008. 14

[323] P. Tan, S. Lin, L. Quan, and H. Y. Shum. Highlight removal by illumination-constrained inpainting. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 164, Washington, DC, USA, 2003. IEEE Computer Society. 23

[324] D. Tao, X. Wang, and W. Bian. Grassmannian regularized structured multi-view embedding for image classification. *IEEE Transactions on Image Processing*, 2013. 5

[325] C. J. Taylor, D. J. Kriegman, and P. Anandan. Structure and motion in two dimensions from multiple images: a least squares approach. In *Visual Motion, 1991. , Proceedings of the IEEE Workshop on*, pages 242–248, oct 1991. 17

[326] C. Tchou, Stumpfel J., P. Einarsson, and M Fajardo. Unlighting the parthenon. In *SIGGRAPH 2004 Sketch*. ACM Press, 2004. 71

[327] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 24(1):52–96, 1983. 14

[328] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:129–139, June 1986. 14

[329] D. Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 417–438, 1988. 14

[330] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool. Towards multi-view object class detection. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1589–1596. IEEE, 2006. 4

[331] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, and L. Van Gool. Using multi-view recognition and meta-data annotation to guide a robot's attention. *The International Journal of Robotics Research*, 28(8):976–998, 2009. 5

[332] R. Timofte, K. Zimmermann, and L. Van Gool. Multi-view traffic sign detection, recognition, and 3d localisation. In *Applications of Computer Vision (WACV), 2009 Workshop on*, pages 1–8. IEEE, 2009. 5

[333] B. Triggs, P. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment–a modern synthesis. *Vision algorithms: theory and practice*, pages 153–177, 2000. 17

[334] E. Trucco and A. Verri. *Introductory Techniques for 3D Computer Vision.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998. 14

[335] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987. 82

[336] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Found. Trends Comput. Graph. Vis.*, 3(3):177–280, 2008. 40

[337] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proceedings of the British Machine Vision Conference*, pages 412–425, 2000. 16

[338] M. Uyttendaele, A. Criminisi, S. B. Kang, S. Winder, R. Szeliski, and R. Hartley. Image-based interactive exploration of real-world environments. *Computer Graphics and Applications, IEEE*, 24(3):52–63, 2004. 5

[339] R. Vaillant and O. D. Faugeras. Using extremal boundaries for 3-d object modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:157–173, February 1992. 19

[340] M. Vergauwen and L. Van Gool. Web-based 3d reconstruction service. *Mach. Vision Appl.*, 17(6):411–426, 2006. 17, 18, 40

[341] L. Vincent. Taking online maps down to street level. *Computer*, 40:118–120, December 2007. 27

[342] ISTI CNR Visual Computing Lab. Gcache, an open source library for cache management. 107

[343] ISTI CNR Visual Computing Lab. Nexus, an open source library for efficient management of 3D data.
`http://vcg.isti.cnr.it/nexus`. 107, 112

[344] ISTI CNR Visual Computing Lab. Photocloud.
`http://vcg.isti.cnr.it/photocloud`. 112

[345] G. Vogiatzis, C. Hernández Esteban, P. H. S. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:2241–2246, December 2007. 19

[346] D. Waltz. Understanding line drawings of scenes with shadows. In Patrick Winston, editor, *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, 1975. 14

[347] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. *ACM SIGGRAPH 2005 Papers on - SIGGRAPH '05*, page 585, 2005. 20

[348] J. Wang, R. Cipolla, and Z. Hongbin. Image-based localization and pose recovery using scale invariant features. In *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*, pages 711–715, 2004. 24

[349] Y. Wang, M. Brookes, and P. L. Dragotti. Object recognition using multi-view imaging. In *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, pages 810–813. IEEE, 2008. 4

[350] K. Welke, J. Issac, D. Schiebener, T. Asfour, and R. Dillmann. Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2012–2019. IEEE, 2010. 5

[351] J. Weng, N. Ahuja, and T. S. Huang. Optimal motion and structure estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 864–884, 1993. 17

[352] A. P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17(1-3):17–45, 1981. 14

[353] A. P. Witkin. Scale-space filtering. In *8th Int. Joint Conf. Artificial Intelligence*, volume 2, pages 1019–1022, Karlsruhe, August 1983. 14

[354] A. P. Witkin, D. Terzopoulos, and M. Kass. Signal matching through scale space. *International Journal of Computer Vision*, 1(2):133–144, 1987. 14

[355] L. B. Wolff. Using polarization to separate reflection components. In *Computer Vision and Pattern Recognition, 1989*, pages 363–369, Jun 1989. 23

[356] L. B. Wolff, S. A. Shafer, and G. E. Healey. *Radiometry. Physics-based vision: principles and practice.* A K Peters/CRC Press, USA, 1992. 14

[357] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle. Surface light fields for 3d photography. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 287–296, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 26

[358] R. J. Woodham. Analysing images of curved surfaces. *Artificial Intelligence*, 17(1-3):117–140, 1981. 14

[359] W. D. Wright. A re-determination of the trichromatic coefficients of the spectral colors. *Transactions of the Optical Society*, 30:141–164, 1928. 34

[360] G. Wyszecki. Correlate for lightness in terms of CIE chromaticity coordinates and luminous reflectance. *Journal of the Optical Society of America*, 57(2):254–254, 1967. 36

[361] Y. Xi and Y. Duan. A integrated depth fusion algorithm for multi-view stereo. *Computer Graphics International*, 2011. 20

[362] J. Xiao, J. Chen, D.-Y. Yeung, and L. Quan. Structuring visual words in 3d for arbitrary-view object localization. In *ECCV '08*, pages 725–737, 2008. 25

[363] W. Xu and J. Mulligan. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 263–270. IEEE, 2010. 5

[364] Z. Xue, J. Yang, Q. Dai, and N. Zhang. Multi-view image denoising based on graphical model of surface patch. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*, pages 1–4. IEEE, 2010. 5

[365] A. Yao, J. Gall, and L. Van Gool. Coupled action recognition and pose estimation from multiple views. *International journal of computer vision*, 100(1):16–37, 2012. 5

[366] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*, pages 239–269. Morgan Kaufmann Publishers Inc., 2003. 14

[367] A. Yezzi and S. Soatto. Stereoscopic segmentation. *International Journal of Computer Vision*, 53(1):31–43, 2003. 19

[368] F. You-jia and L. Jian-wei. Rotation invariant multi-view color face detection based on skin color and adaboost algorithm. In *Biomedical Engineering and Computer Science (ICBECS), 2010 International Conference on*, pages 1–5. IEEE, 2010. 4

[369] L. Zhang, S. Vaddadi, H. Jin, and S. K. Nayar. Multiple view image denoising. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1542–1549. IEEE, 2009. 5

[370] W. Zhang and J. Kosecka. Image based localization in urban environments. *3DPVT'06*, pages 33–40, Jun 2006. 24

[371] J. Y. Zheng. Acquiring 3-d models from sequences of contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:163–178, 1994. 19

[372] L. Zhu, Y. Chen, A. Torralba, W. Freeman, and A. Yuille. Part and appearance sharing: Recursive compositional models for multi-view. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1919–1926. IEEE, 2010. 4

[373] X. Zhu, P. Zhang, J. Shao, Y. Cheng, Y. Zhang, and J. Bai. A snake-based method for segmentation of intravascular ultrasound images and its in vivo validation. *Ultrasonics*, 51(2):181–189, 2011. 5

[374] Z. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H. S. Sawhney. Real-time global localization with a pre-built visual landmark database. In *CVPR*, pages 1–8, 2008. 25

[375] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin. Environment matting and compositing. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 205–214. ACM Press/Addison-Wesley Publishing Co., 1999. 26