

Shape from photographs: a multi-view stereo pipeline

Carlos Hernández and George Vogiatzis

Abstract Acquiring 3d shape from images is a classic problem in Computer Vision occupying researchers for at least 20 years. Only recently however have these ideas matured enough to provide highly accurate results. We present a complete algorithm to reconstruct 3d objects from images using the stereo correspondence cue. The technique can be described as a pipeline of four basic building blocks: camera calibration, image segmentation, photo-consistency estimation from images, and surface extraction from photo-consistency. In this chapter we will put more emphasis on the latter two: namely how to extract geometric information from a set of photographs without explicit camera visibility, and how to combine different geometry estimates in an optimal way.

1 Introduction

Digital modeling of 3d scenes is becoming increasingly popular and necessary for a wide range of applications such as cultural heritage preservation, online shopping or computer games. Although active methods [34, 49] remain one of the most popular techniques of acquiring shape, the high cost of the equipment, complexity, and difficulties to capture color are three big disadvantages. As opposed to active techniques, photograph-based techniques provide an efficient and easy way to acquire shape and color by simply capturing a sequence of photographs of the object.

The goal of any shape-from-photographs algorithm can be described as “*given a set of input photographs, how to estimate a 3d shape that would generate the same photographs, assuming same material, viewpoints and lighting conditions*”. This definition highlights the main difficulty of the problem: photographs are obtained

Carlos Hernández
Toshiba Research Cambridge, e-mail: carlos.hernandez@crl.toshiba.co.uk

George Vogiatzis
Toshiba Research Cambridge, e-mail: george.vogiatzis@crl.toshiba.co.uk

as a result of complex interactions between the geometry of the scene, the materials of the scene, the lighting conditions and the viewpoints (see Fig. 1). Hence

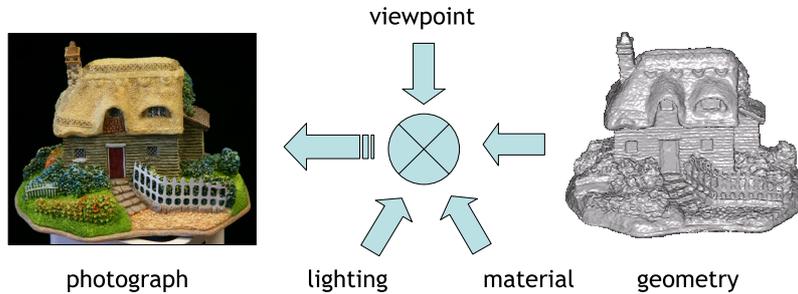


Fig. 1 Image formation model. The image of a 3d scene depends on its geometry, material properties, lighting conditions and pose of the viewer.

recovering the geometry just from photographs is not only a challenging problem but also, in the general case, an ill-posed problem. It is challenging because lighting and material properties play a very important role in the image formation model. The same geometry with a different material or different lighting conditions can give extremely different photographs. It is also an ill-posed problem because, in the general case, different combinations of geometry, lighting and material can produce exactly the same photographs, making it impossible to recover a single scene geometry. The main recipe to make the problem well-posed is to use priors on the types of surface that one expects. Traditionally the most common type of prior is the smooth surface prior. However when dealing with special classes of objects such as human faces or man-made objects, more evolved priors have been successfully used, *e.g.* human faces [54], buildings [53] or planes [15].

As for the importance of materials and lighting conditions, it has been addressed by restricting the class of materials a particular algorithm is designed for. As a result, no single method is able to correctly reconstruct a general scene with any type of materials and lighting conditions, leading to a plethora of specific algorithms designed for specific types of objects and using specific cues: silhouettes [1], texture [50], transparency [44], defocus [14], shading [51] or correspondence, both sparse [3] and dense [40]. Historically the most successful cues have been silhouettes, correspondence, and shading. Silhouettes and correspondences are the most robust of all due to their invariance to illumination changes. The shading cue needs a more controlled illumination environment, but it can produce breathtaking results, which makes it widely used too. An example of an algorithm [23] exploiting the shading cue is shown in Fig. 2. The algorithm is designed to find a 3d shape that produces the same shading as the original object. Interestingly, if the estimated 3d shape is then used to manufacture a replica from a different material (in Fig. 2 the original is porcelain, while the replica is plaster) we can appreciate how the replica still



Fig. 2 Shading comparison of a porcelain figurine and a manufactured replica obtained using [23]. The original porcelain figurine is shown on the left, while a manufactured replica using the 3d model obtained using [23] is shown on the right. The material of the replica is plaster. See how the replica perfectly imitates the shading component, even though the materials are different.

shows the same shading pattern. This is the desired behavior, since the algorithm is specifically designed to imitate the shading, not to produce identical photographs.

Among the vast literature available on image-based modeling techniques, recent work on multi-view stereo (MVS) reconstruction has become a growing area of interest in recent years with many differing techniques achieving a high degree of accuracy [40]. These techniques are mainly based on the correspondence cue and focus on producing 3d models from a sequence of calibrated images of an object, where the intrinsic parameters and pose of the camera are known. In addition to providing a taxonomy of methods, [40] also provides a quantitative analysis of performance both in terms of accuracy and completeness. If we take a look at the top performers, they may be loosely divided into two groups. The first group make use of techniques such as correspondence estimation, local region growing and filtering to build up a “*cloud of patches*” [17, 19, 35, 36] that can be optionally made dense using meshing algorithms such as Poisson reconstruction [4] or signed distance functions [12]. The second group make use of some form of global optimization strategy on a volumetric representation to extract a surface [18, 20, 24, 47, 48]. Under this second paradigm, a 3d cost volume is computed, and then a 3d surface is extracted using tools previously developed for the 3d segmentation problem such as deformable models [20], level-sets [13, 39] or graph-cuts [6, 33, 16, 24, 41, 46, 48].

The way volumetric methods usually exploit photo-consistency is by building a 3d map of photo-consistency where each 3d location gives an estimate of how photo-consistent would be the reconstructed surface at that location. The only requirement to compute this photo-consistency 3d map is that camera visibility is available. Unfortunately, the geometry of the scene, *i.e.* what we try to compute, is required to know which cameras see a 3d location (see Fig. 3). In order to break

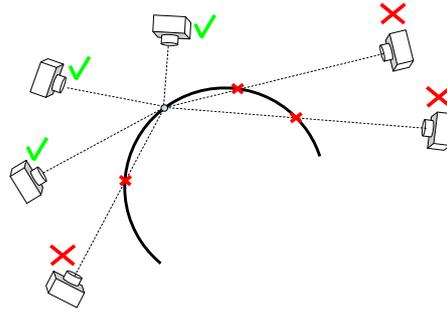


Fig. 3 Occlusion problem. In order to compute shape using photo-consistency, the camera visibility is required. At the same time, in order to compute the camera visibility, the shape is required.

this dependency between visibility and shape, multi-view stereo algorithms have taken different approaches. A majority of methods use the notion of “*current surface*” in order to jointly optimize for camera visibility and shape. The visibility computed from the reconstructed surface at iteration $i - 1$ is then used to compute photo-consistency at iteration i , improving the reconstruction gradually [13]. Some methods use a proxy of the true surface to estimate visibility, such as the visual hull [24, 48]. Finally, a third category of methods try to compute a “*visibility-independent*” photo-consistency where occlusion is treated as an additional source of image noise [7, 18, 20].

In this chapter we will give further insight into a two-stage MVS volumetric approach: namely how to extract a 3d volume of photo-consistency from a set of photographs without explicit camera visibility in section 3, and how to extract a surface from the photo-consistency volume in a globally optimal way in section 4. The pipeline described in this chapter is currently a top performer in the recent evaluation of multi-view stereo algorithms by Seitz et al[40].

2 Multi-view stereo pipeline: from photographs to 3d models

There exists a vast literature on multi-view stereo algorithms. Even though many of the methods share the same basic architecture, they differ mainly in what type of scenes or computation time they are optimized to work with. All the multi-view stereo methods use the correspondence cue, which is usually exploited in the form of a photo-consistency metric such as Normalized Cross Correlation, Sum of Square Differences, or Mutual Information. Starting from the photo-consistency metric, different algorithms focus on different target applications such as outdoor scenes [45], building reconstruction [11, 37, 38], interior buildings [15] or object reconstruction [40]. In this chapter we describe a volumetric multi-view stereo approach that is optimized for general scene reconstruction, with a preference for watertight surfaces. The pipeline (see figure 4) can be described as:

- photograph acquisition,
- camera calibration,
- computing 3d photo-consistency from a set of calibrated photographs,
- extracting a 3d surface from a 3d map of photo-consistency.

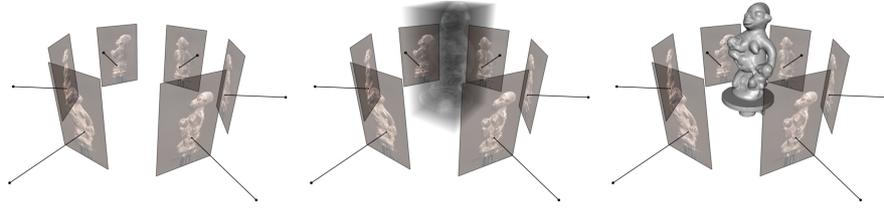


Fig. 4 3d multi-view stereo pipeline. Image calibration, photo-consistency 3d map from a set of photographs (section 3) and surface extraction from a photo-consistency 3d map (section 4).

In the following sections we focus on how to extract 3d photo-consistency from a set of photographs (see section 3) and how to use the 3d photo-consistency to extract a 3d surface (see section 4). We leave the discussion on image acquisition, *e.g.* real-time vs photograph-based, and on camera calibration for future discussion (see [43] for an state-of-the-art system to calibrate a set of photographs).

3 Computing photo-consistency from a set of calibrated photographs

Given a set of images and their corresponding camera poses, we would like to extract a 3d map of photo-consistency that tell us how photo-consistent is a particular 3d location **for a given set of visible cameras**. The main difficulty of this step is how to produce a volumetric measure of photo-consistency without the knowledge of the set of cameras that should be used to compute photo-consistency for every 3d location.

This problem is addressed in the proposed 3d modeling pipeline by following a similar approach to [20] where photo-consistency is made robust to occlusion. This approach computes a 3d map of photo-consistency as an aggregation of depth-maps from different view-points (see Fig. 5). The creation of such a photo-consistency 3d map is similar in spirit to the space carving approach proposed by [32]. However, by computing it as an aggregation of depth-maps, two advantages appear:

- depth-map computation using dense stereo is a very successful and active research topic. It is an ideal building block to use since improvements in the field of dense stereo can be directly beneficial to the multi-view stereo problem.

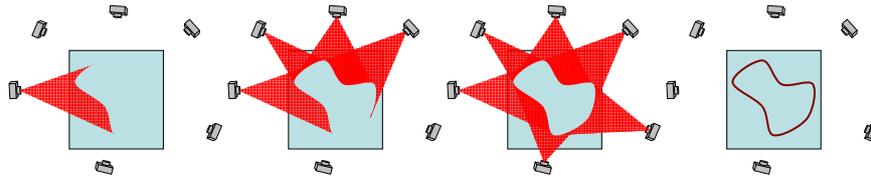


Fig. 5 Computing a photo-consistency volume as aggregation of depth-maps. From left to right, three different stages of merging individual depth-maps into a single photo-consistency volume. Right shows the final photo-consistency volume.

- Computation time is no longer dependent on the resolution of the 3d volume, but on the number of cameras. It is also highly parallelizable, since each depth-map is independently computed and no iterated visibility computation is required.

By building a 3d map of photo-consistency, the 3d reconstruction problem can now be seen as a 3d segmentation task, allowing us to use algorithms previously developed for 3d segmentation. These algorithms include deformable surfaces [20], Poisson reconstruction [17], signed distance functions [18], Delaunay [7] or MRFs [22, 29, 47].

A comparison of the importance of this stage in the reconstruction pipeline is shown in Fig. 6. The occlusion-robust photo-consistency of [20] (Fig. 6 middle)

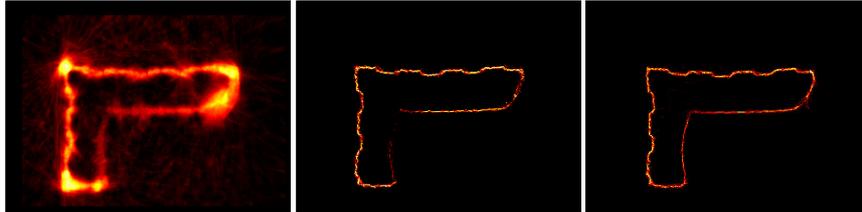


Fig. 6 Noise reduction in photo-consistency. Left: a slice of the photo-consistency used in [48] contains falsely photo-consistent regions (*e.g.* near the corners). Middle: occlusion robust photo-consistency proposed in [20] significantly suppresses noise and the correct surface can be accurately localized. One side of the vertical wall is missing due to heavy occlusions. Right: occlusion robust photo-consistency proposed in [8]. The vertical wall is correctly represented.

clearly outperforms [48] (Fig. 6 left). However, since this method exploits the redundancy between images to be robust against occlusion, it suffers with sparse data sets (see the missing vertical wall in Fig. 6 middle). An improved version of the occlusion-robust photo-consistency has been proposed in [8] that is capable of better dealing with sparse data sets (see the improvement in the vertical wall in Fig. 6 right). We adopt [8] in our multi-view stereo pipeline as the building block to compute individual depth-maps. In the remaining of this section we describe this algorithm more in detail.

3.1 Normalized Cross Correlation for depth-map computation

Normalized Cross Correlation (NCC) may be used to define an error metric for matching two windows in different images. Figure 7 provides an example of using NCC and epipolar geometry to perform window based matching. If we fix a pixel location in a reference image, for each possible depth away from that pixel we get a corresponding pixel in the second image. By computing the NCC between windows centered in those two pixels we can define a matching score as a function of depth for the reference pixel. We refer to this function as the *correlation curve* of the pixel. A typical correlation curve will exhibit a very sharp peak at the correct depth, and possibly a number of secondary peaks in other depths.

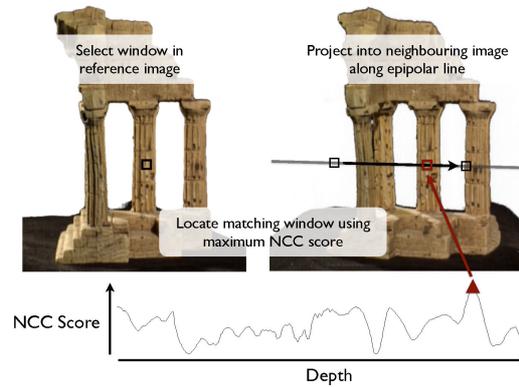


Fig. 7 Normalized Cross-Correlation based window matching.

In [20] a depth-map is generated for each input image using this matching technique for neighboring images. For each pixel a number of correlation curves are computed (using a few of the neighboring viewpoints) and the depth that gives rise to most peaks in those curves is selected as the depth for that pixel. See [20] or [47] for details. This process results in an independent depth estimate for each pixel. These depth estimates will unavoidably contain a significant percentage of outliers which must be dealt with in the subsequent step of [20] which is the volumetric fusion of multiple depth-maps. In data sets with a large number of images this is overcome by the redundancy in the depth-estimates. The same surface point is expected to be covered by many different depth-maps, some of which will have the right depth estimate. In sparse data-sets however, each surface point may be seen by as few as two or three depth-maps. It is therefore crucial that outliers are minimized in the depth-map generation stage.

In order to efficiently exploit NCC as a photo-consistency measure, we need to focus on the two most significant failure modes of NCC matching which are (1) the presence of repetitions in the texture and (2) complete matching failure due to occlusion, distortion and lack of texture. These are now described in more detail.

3.1.1 Repeating texture

In general, there is no guarantee that the appearance of a patch is unique across the surface of the object. This results in correlation curve peaks at incorrect depths due to repeated texture — ‘false’ matches (Fig. 7). A larger window size is more likely to uniquely match to the true surface, reducing the number of false matches. However the associated peak will be broader and less well localized, reducing the accuracy of the depth estimate. The absolute value of the NCC score at a peak reflects how well the two windows match. Thus one might expect the peak with the maximum score to be the true peak. Unfortunately, the appearance of false matches due to repeated texture may result in false peaks having similar or even greater scores than the true surface peak (Fig. 8 (a)). To identify the correct peak, we propose to apply a spatial consistency constraint across neighboring pixels in the depth-map. The underlying assumption is that if a peak corresponds to the true surface, the neighboring pixels should have peaks at a similar depth. The exception to this is occlusion boundaries, which are however catered for under the next failure mode.

3.1.2 Matching failure

The second failure mode is comprised of occlusion errors, distorted image windows (due to slanted surfaces) and lack of texture. In all of these cases, the correlation curve will not exhibit a peak at the true depth of the surface, resulting in only false peaks. Furthermore no spatial consistency can be enforced between the pixel in question and its neighbors. In this situation we would like to acknowledge that the depth at this pixel is unknown and should therefore offer no vote for the surface location.

In order to achieve these two goals we propose an optimization strategy which makes use of a discrete label Markov Random Field (MRF). The MRF allows each pixel to choose a depth corresponding to one of the top NCC peaks which is spatially consistent with neighboring pixels or select an *unknown* label to indicate that no such peak occurs and there is no correct depth estimate. This process means that the returned depth map should only contain accurate depths, estimated with a high degree of certainty, and an *unknown* label for pixels which have no certain associated depth. Figure 8 illustrates the optimization for a 1D example of neighboring pixels across an occlusion boundary.

3.2 Depth Map Estimation

The proposed algorithm estimates the depth for each pixel in the input images. It proceeds in two stages: Initially we extract a set of possible depth values for each pixel using NCC as a matching metric. We then solve a multi-label discrete MRF model which yields the depth assignment for every pixel. One of the key features

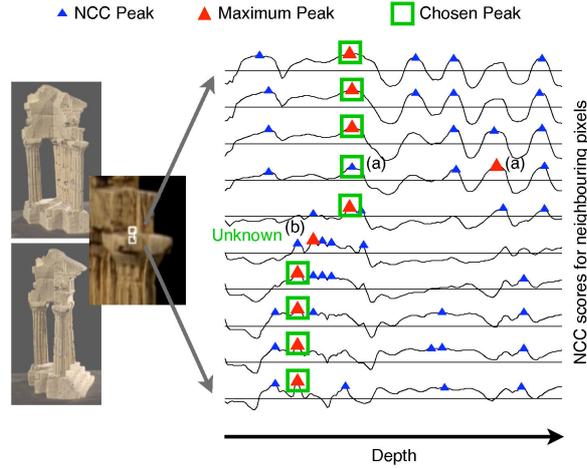


Fig. 8 Illustration of the MRF optimization applied to neighboring pixels. Existing method return the maximum peak which results in outliers in the depth estimate. The MRF optimization corrects an outlier to the true surface peak (a) and introduces an unknown label at the occlusion boundary (b)

in this process is the inclusion of an *unknown* state in the MRF model. This state is selected when there is insufficient evidence for the correct depth to be found.

3.2.1 Candidate Depths

The input to the algorithm is a set of calibrated images \mathcal{I} and the output is a set of corresponding depth-maps \mathcal{D} . In the following, we describe how to acquire a depth-map for a reference image $I_{\text{ref}} \in \mathcal{I}$. Let $N(I_{\text{ref}})$ denote a set of ‘neighboring’ images to I_{ref} .

As proposed in section 3.1, we wish to obtain a hypothesis set of possible depths for each pixel $p_i \in I_{\text{ref}}$. Taking each pixel in turn, we project the epipolar ray into a second image $I_n \in I_{\text{ref}}$ and sample the NCC matching score over a depth range $\rho_i(z)$. We compute the score using a rectangular window centered at the projected image co-ordinates. One of the advantages of the multiple depth hypotheses is the ability to use a smaller matching window to provide a faster computation and improved localization of the surface. Once we have obtained the sampled ray we store the top K peaks $\hat{\rho}_i(z_{i,k}), k \in [1, K]$ with the greatest NCC score for each pixel. Depending on the number of images available, and the width of the camera baseline, this process may be repeated for other neighboring images. We then continue to the optimization stage with a set of the best K possible depths, and their corresponding NCC scores, over all neighboring images of I_{ref} .

3.2.2 MRF Formulation

At this stage a set of candidate depths $\hat{\rho}_i(z_{i,k}), k \in [1, K]$, for each pixel p_i in the reference image I_{ref} has been assigned and we wish to determine the correct depth map label for each pixel. As described in section 3.1, we also make use of an *unknown* state to account for the failure modes of NCC matching.

We model the problem as a discrete MRF where each pixel has a set of up to $(K + 1)$ labels. The first K labels, fewer if an insufficient number of peaks were found during the matching stage, correspond to the peaks in the NCC function and have associated depths $z_{i,k} \in \mathcal{Z}_i$ and scores $\hat{\rho}_i(z_{i,k})$. The final state is the *unknown* state \mathcal{U} . If the optimization returns this state, the pixel is not assigned a depth in the final depth map. For each pixel we therefore form an augmented label set $z'_{i,k} \in \{\mathcal{Z}_i, \mathcal{U}\}$ to include the unknown state.

The optimization assigns a label $\bar{k}_i \in \{1 \dots K, \mathcal{U}\}$ to each pixel p_i . The cost function to be minimized consists of unary potentials for each pixel and pairwise interactions over first order cliques. The cost of a labeling $\bar{\mathbf{k}} = \{\bar{k}_i\}$ is expressed as

$$E(\bar{\mathbf{k}}) = \sum_i \phi(\bar{k}_i) + \sum_{(i,j)} \psi(\bar{k}_i, \bar{k}_j) \quad (1)$$

where i denotes a pixel and (i, j) denote neighboring pixels.

The following sections discuss the formulation of the unary potentials $\phi(\cdot)$ and pairwise interactions $\psi(\cdot, \cdot)$.

3.2.3 Unary Potentials

The unary labeling cost is derived from the NCC score of the peak. We wish to penalize peaks with a lower matching score since they are more likely to correspond to an incorrect match due to occlusion or noise. The NCC process will always return a score in the range $[-1, 1]$. As is common practice, [47], we take an inverse exponential function to map this score to a positive cost.

The unary cost for the *unknown* state is set to a constant value $\phi_{\mathcal{U}}$. This term serves two purposes. Firstly it acts as a cut-off threshold for peaks with poor NCC scores which have no pairwise support (neighboring peaks of similar depth). This mostly accounts for peaks which are weakly matched due to distortion or noise. Secondly it acts as a truncation on the depth disparity cost of the pairwise term. By assigning a low pairwise cost between peaks and the *unknown* state, the constant unary cost will effectively act as a threshold on the depth disparity to handle the case of an occlusion boundary. Thus the final unary term is given by

$$\phi(k_i = x) = \begin{cases} \lambda e^{-\beta \hat{\rho}_i(z_{i,x})} & x \in [1 \dots K] \\ \phi_{\mathcal{U}} & x = \mathcal{U} \end{cases} . \quad (2)$$

3.2.4 Pairwise Interactions

The pairwise labeling cost is derived from the disparity in depths of neighboring peaks. As has been previously mentioned, this term is not intended to provide a strong regularization of the depth map. Instead it is used to try and determine the correct peak, corresponding to the true surface location, out of the returned peaks. We observe that the correct peak may not have the maximum score. Therefore if there is strong agreement on depth between neighboring peaks, we take this to be the true location of the surface.

When dealing with the depth disparity term we are really considering surface orientation; whether the surface normal is pointing towards or away from the camera. Under a perspective projection camera model it is therefore necessary to correct for the absolute depth of the peaks rather than simply taking the difference in depth. We perform this correction by dividing by the average depth of the two peaks. The resulting pairwise term is given by

$$\psi(k_i = x, k_j = y) = \begin{cases} 2 \frac{|z_{i,x} - z_{j,y}|}{(z_{i,x} + z_{j,y})} & x \in [1 \dots K] \quad y \in [1 \dots K] \\ \psi_U & x = \mathcal{U} \quad y \in [1 \dots K] \\ \psi_U & x \in [1 \dots K] \quad y = \mathcal{U} \\ 0 & x = \mathcal{U} \quad y = \mathcal{U} \end{cases} . \quad (3)$$

We set ψ_U to a small value to encourage regions with many pixels labeled as *unknown* to coalesce. This acts as a further stage of noise reduction since it prevents spurious peaks with high scores but no surrounding support from appearing in regions of occlusion.

3.2.5 Optimization

To obtain the final depth map we need to determine the optimal labeling $\hat{\mathbf{k}}$ such that

$$E(\hat{\mathbf{k}}) = \arg \min_{(\mathbf{k})} \sum_i \phi(\bar{k}_i) + \sum_{(i,j)} \psi(\bar{k}_i, \bar{k}_j) . \quad (4)$$

Since in the general case this is an NP-hard problem we must use an approximate minimization algorithm to achieve a solution. The most well-known techniques for solving problems of this nature are based on graph-cuts and belief propagation. Instead, we use the recently developed sequential tree-reweighted message passing algorithm, termed TRW-S, of [30]. This has been shown to outperform belief propagation and graph-cuts in tests on stereo matching using a discrete number of disparity levels. In addition to minimizing the energy, the algorithm estimates a lower

bound on the energy at each iteration which is useful in checking for convergence and evaluating the performance of the algorithm. We should note, however, that we are by no means guaranteed that the lower bound is attainable.

3.3 Photo-consistency 3d map from a set of depth-maps

In order to create a 3d volume of photo-consistency from a set of depth-maps \mathcal{D} we “*uplift*” every depth-map in \mathcal{D} into 3d using the camera calibration data. The photo-consistency of a 3d point \mathbf{x} is defined as the sum of the confidences of all its nearby depth-map points. That is, given all the uplifted depth-map 3d points \mathbf{d}_i and their corresponding confidence values s_i , the photo-consistency $\mathcal{C}(\mathbf{x})$ can be define as

$$\mathcal{C}(\mathbf{x}) = \sum_{i:|\mathbf{x}-\mathbf{d}_i|<\varepsilon} s_i, \quad (5)$$

where ε is a pre-defined ball size. If the photo-consistency is to be discretize using a volumetric grid, then ε is simply the size of a voxel.

4 Extracting a 3d surface from a 3d map of photo-consistency

Given a 3d map of photo-consistency, we would like to extract a 3d surface. As mentioned earlier, by building a 3d map of photo-consistency, the reconstruction problem can now be solved using 3d segmentation techniques. Out of all the segmentation algorithms available, MRF approaches are very widely spread due to its global convergence properties. They also allow the fusion of different cues in an elegant way, *e.g.* see [29]. One of the main criticisms of MRFs applied to 3d segmentation is the discretization artifacts originating from its discrete nature. In order to remove them, the surface is usually further refined using a continuous formulation such as level-sets [13, 39] or deformable models [20], allowing for a finer control of the regularization than the one provided by MRFs. In the remaining of this section we describe the MRF framework for multi-view stereo first proposed by [47] and further extended in [22]. We also describe the deformable model by [20] that we use as a refinement step.

4.1 Multi-view stereo using multi-resolution graph-cuts

In [5] and subsequently in [2] it was shown how graph-cuts can optimally partition 2d or 3d space into ‘foreground’ and ‘background’ regions under any cost functional consisting of the following two terms:

- **Labeling cost or unary term:** for every point in space there is a cost for it being labeled ‘foreground’ or ‘background’.
- **Discontinuity cost or binary term:** for every point in space, there is a cost for it lying on the boundary between the two partitions.

Mathematically, the cost functional described above can be seen as the sum of a weighted *surface area* of the boundary surface and a weighted *volume* of the ‘foreground’ region as follows:

$$E(S) = \int_S \rho(\mathbf{x}) dA + \int_{V(S)} \sigma(\mathbf{x}) dV \quad (6)$$

where S is the boundary between ‘foreground’ and ‘background’, $V(S)$ denotes the ‘foreground’ volume enclosed by S and ρ and σ are two scalar density fields. The application described in [5] was the problem of 2d/3d segmentation. In that domain $\rho(\mathbf{x})$ is defined as a function of the image intensity gradient and $\sigma(\mathbf{x})$ as a function of the image intensity itself or local image statistics.

In the framework of the multi-view stereo problem, this model balances two competing terms: the first one minimizes a surface integral of photo-consistency (binary term) while the second one maximizes the volume of regions with a high evidence of being foreground (unary term). In the literature, it is usually the binary term that is data driven, while the unary term is just used as a basic prior, *e.g.* as a ballooning term [9]. In this work, we use the photo-consistency 3d map computed in section 3 as the binary term. As for the unary term, very little work has been done to obtain an appropriate ballooning term. In most of the previous work on volumetric multi-view stereo the ballooning term is a very simplistic inflationary force that is constant in the entire volume, *i.e.* $\sigma(\mathbf{x}) = -\lambda$. This simple model tries to recover thin structures by maximizing the volume inside the final surface. However, as a side effect, it also fills in concavities behaving as a regularization force and smoothing fine details.

When silhouettes of the object are available, an additional *silhouette cue* can be used [24, 48], which provides the constraint that all points *inside* the object volume must project inside the silhouettes of the object. Hence the silhouette cue can provide some foreground/background information by giving a very high likelihood of being *outside* the object to 3d points that project outside the silhouettes. However this ballooning term is not enough if thin structures or big concavities are present, in which case the method fails (see Fig. 16 middle row). Very recently, a data driven, foreground/background model based on the concept of *photo-flux* has been introduced [6]. However, the approach requires approximate knowledge of the object surface orientation which in many cases is not readily available.

Ideally, the ballooning term should be linked to the notion of visibility, where points that are not visible from any camera are considered to be inside the object or **foreground**, and points that are visible from at least one camera are considered to be outside the object or **background**. An intuition of how to obtain such a ballooning term is found in a classic paper on depth sensor fusion by Curless and Levoy [12]. In that paper the authors fuse a set of depth sensors using signed distance functions. This fusion relies on the basic principle that the space between the sensor and

the depth map should be empty or background, and the space after the depth map should be considered as foreground. In this section we follow the approach by [22] where this visibility principle is generalized and computed in a probabilistic version by calculating the “*evidence of visibility*” from a given set of depth-maps. The “*evidence of visibility*” is then used as an intelligent ballooning term.

The outline of the full system is as follows:

- create a set of depth-maps from the set of input calibrated photographs,
- compute the photo-consistency 3d map from the set of depth-maps,
- derive the discontinuity cost $\rho(\mathbf{x})$ from the photo-consistency 3d map,
- derive the labeling cost $\sigma(\mathbf{x})$ from the set of depth-maps, *i.e.* use a data-aware ballooning term computed from the evidence of visibility and,
- extract the final surface as the global solution of the min-cut problem given $\rho(\mathbf{x})$ and $\sigma(\mathbf{x})$.

A real example of discontinuity and labeling costs is shown in Fig.9. Note they have been computed on a multi-resolution grid.

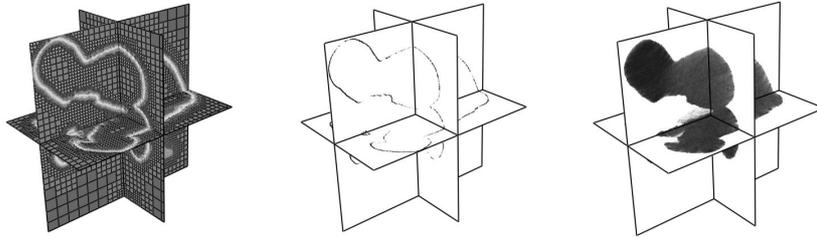


Fig. 9 Different terms used in the graph-cut algorithm to reconstruct the Gormley sculpture of Fig. 16. Left: multi-resolution grid used in the graph-cut algorithm. Middle: Discontinuity cost $\rho(\mathbf{x})$ (or photo-consistency). Right: labeling cost $\sigma(\mathbf{x})$ (or intelligent ballooning).

The algorithm just described can also be used when the input is no longer a set of photographs but a set of depth-maps obtained from other types of sensor, *e.g.* laser scanner. In this case, the system just skips the first step, since the depth-maps are already available, and computes ρ and σ directly from the set of depth-maps given as input.

4.2 Discontinuity cost from a set of depth-maps

Once we have computed a depth-map for every input image, we can build the photo-consistency 3d map $\rho(\mathbf{x})$ for every 3d location \mathbf{x} as explained in section 3.3. Since the graph-cut algorithm **minimizes** the discontinuity cost, and we want to *maximize* the photo-consistency, we invert the discontinuity map $\rho(\mathbf{x})$ using the exponential:

$$\rho(\mathbf{x}) = e^{-\mu\mathcal{C}(\mathbf{x})}, \quad (7)$$

where μ is a very stable rate-of-decay parameter that converts photo-consistency scores into a normalized discontinuity cost in the range $[0, 1]$.

As a way of improving the big memory requirements of graph-cut methods, we propose to store the values of $\rho(\mathbf{x})$ in an octree partition of 3d space. The size of the octree voxel will depend on the photo-consistency value $\mathcal{C}(\mathbf{x})$. Voxels with a non-zero photo-consistency value will have the finest resolution while the remaining space where $\mathcal{C}(\mathbf{x}) = 0$ will be partitioned using bigger voxels, the voxel size being directly linked with the distance to the closest non-empty voxel (see Fig. 9 for an example of such an octree partition).

4.3 Graph structure

To obtain a discrete solution to Equation (6) 3d space is quantized into voxels using an octree partition. The graph nodes consist of all voxels whose centers are within a certain bounding box that is guaranteed to contain the object. For the results presented here these nodes were connected with a regular 6-neighborhood grid. Bigger neighborhood systems can be used which provide a better approximation to the continuous functional (6), at the expense of using more memory to store the graph. Now assume two voxels centered at \mathbf{x}_i and \mathbf{x}_j are neighbors. Let the smaller voxel be size $h \times h \times h$. Then the weight of the edge joining the two corresponding nodes on the graph will be [5]

$$w_{ij} = \frac{4\pi h^2}{3} \rho\left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}\right) \quad (8)$$

where $\rho(\mathbf{x})$ is the matching cost function defined in (7). In addition to these weights between neighboring voxels there is also the ballooning force edge connecting every voxel to the source node with a constant weight of $w_b = \lambda h^3$. Finally, the outer voxels that are part of the bounding box (or the voxels outside the visual hull if that is available) are connected with the sink with edges of infinite weight. The configuration of the graph is shown in figure 10 (right).

It is worth pointing out that the graph structure described above can be thought of as a simple binary MRF. Variables correspond to voxels and can be labeled as being *inside* or *outside* the scene. The unitary clique potential is just 0 if the voxel is outside and w_b if it is inside the scene while the pairwise potential between two neighbor voxels i and j is equal to w_{ij} if the voxels have opposite labels and 0 otherwise. As a binary MRF with a *sub-modular* energy function [31] it can be solved exactly in polynomial time using Graph-cuts.

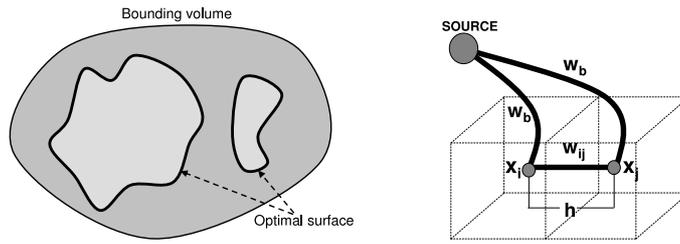


Fig. 10 Surface geometry and flow graph construction. On the left: a 2d slice of space showing the bounding volume and the optimal surface inside it that is obtained by computing the minimum cut of a weighted graph. Note that complicated topologies such as holes or disjoint volumes can be represented by the model and recovered after optimization. On the right: the correspondence of voxels with nodes in the graph. Each voxel is connected to its neighbors as well as to the source.

4.4 Labeling cost from a set of depth-maps

In the same way as the computation of the discontinuity cost, the ballooning term $\sigma(\mathbf{x})$ can be computed exclusively from a set of depth-maps. We propose to use the probabilistic evidence for visibility proposed by [22] and described in section 4.5 as an *intelligent* ballooning term. To do so, all we need is to choose a noise model for the sensor given a depth-map D and its confidence $\mathcal{C}(D)$. We propose to use a simplistic yet powerful model of a Gaussian contaminated with a uniform distribution, *i.e.* an inlier model plus an outlier model. The inlier model is assumed to be a Gaussian distribution centered around the true depth. The standard deviation is considered to be a constant value that only depends on the image resolution and camera baseline. The outlier ratio varies according to the confidence on the depth estimation $\mathcal{C}(D)$, and in this work is just proportional to it. The labeling cost $\sigma(\mathbf{x})$ at a given location is just the evidence of visibility. The details of this calculation are laid out in the next section.

4.5 Probabilistic fusion of depth sensors

This section considers the problem of probabilistically fusing depth maps obtained from N depth sensors. We will be using the following notation: The sensor data is a set of N depth maps $\mathcal{D} = D_1, \dots, D_N$. A 3d point \mathbf{x} can therefore be projected to a pixel of the depth map of the i -th sensor and the corresponding depth measurement at that pixel is written as $D_i(\mathbf{x})$ while $D_i^*(\mathbf{x})$ denotes the true depth of the 3d scene. The measurement $D_i(\mathbf{x})$ contains some noise which is modeled probabilistically by a pdf conditional on the real surface depth

$$p(D_i(\mathbf{x}) | D_i^*(\mathbf{x})). \quad (9)$$

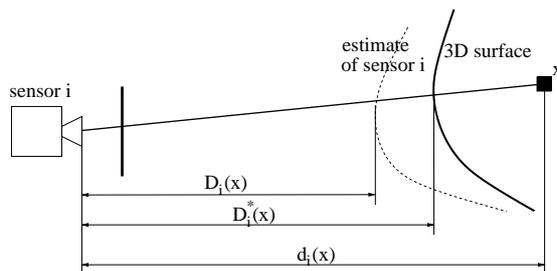


Fig. 11 Sensor depth notation. Sensor i measures the depth of the scene along the optic ray from the sensor to 3d point \mathbf{x} . The depth of point \mathbf{x} from sensor i is $d_i(\mathbf{x})$ while the correct depth of the scene along that ray is $D_i^*(\mathbf{x})$ and the sensor measurement is $D_i(\mathbf{x})$.

The depth of the point \mathbf{x} away from the sensor is $d_i(\mathbf{x})$ (see figure 11). If \mathbf{x} is located on the 3d scene surface then $\forall i D_i^*(\mathbf{x}) = d_i(\mathbf{x})$. If for a particular sensor i we have $D_i^*(\mathbf{x}) > d_i(\mathbf{x})$ this means that the sensor can *see beyond* \mathbf{x} or in other words that \mathbf{x} is *visible* from the sensor. We denote this event by $V_i(\mathbf{x})$. When the opposite event $\bar{V}_i(\mathbf{x})$ is true, as in figure 11, then \mathbf{x} is said to be *occluded* from the sensor. To fuse these measurements we consider a predicate $V(\mathbf{x})$ which is read as: ' \mathbf{x} is visible from at least one sensor'. More formally the predicate is defined as follows:

$$V(\mathbf{x}) \equiv \exists i V_i(\mathbf{x}) \quad (10)$$

$V(\mathbf{x})$ acts as a proxy for the predicate we should ideally be examining which is ' \mathbf{x} is outside the volume of the 3d scene'. However the sensors cannot provide any evidence beyond $D_i^*(\mathbf{x})$ along the optic ray, the rest of the points on that ray being occluded. If there are locations that are occluded from all sensors, no algorithm could produce any evidence for these locations being inside or outside the volume. In that sense therefore, $V(\mathbf{x})$ is the strongest predicate one could hope for in an optical system. An intuitive assumption made throughout this section is that the probability of $V(\mathbf{x})$ depends only on the depth measurements of sensors along optic rays that go through \mathbf{x} . This means that most of the inference equations will be referring to a single point \mathbf{x} , in which case the \mathbf{x} argument can be safely removed from the predicates.

The set of assumptions which we denote by \mathcal{J} consists of the following:

- The probability distributions of the true depths of the scene $D_1^*(\mathbf{x}) \cdots D_N^*(\mathbf{x})$ and also of the measurements $D_1(\mathbf{x}) \cdots D_N(\mathbf{x})$ are independent given \mathcal{J} (see figure 12 for justification).
- The probability distribution of of a sensor measurement given the scene depths and all other measurements only depends on the surface depth it is measuring:

$$p(D_i | D_1^* \cdots D_N^* D_{j \neq i} \mathcal{J}) = p(D_i | D_i^* \mathcal{J}) \quad (11)$$

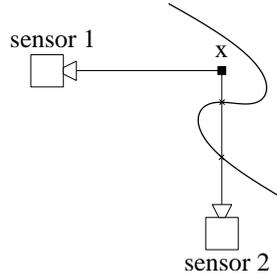


Fig. 12 Visibility from sensors. In the example shown above the point is not visible from sensor 2 while it is visible from sensor 1, *i.e.* we have $V_1\bar{V}_2$. In the absence a surface prior that does not favor geometries such as the one shown above, one can safely assume that there is no probabilistic dependence between visibility or invisibility from any two sensors.

We are interested in computing the evidence function under this set of independence assumptions [26] for the visibility of the point given all the sensor measurements:

$$e(V | D_1 \cdots D_N \mathcal{J}) = \log \frac{p(V | D_1 \cdots D_N \mathcal{J})}{p(\bar{V} | D_1 \cdots D_N \mathcal{J})}. \quad (12)$$

From \mathcal{J} and rules of probability one can derive:

$$p(\bar{V} | D_1 \cdots D_N \mathcal{J}) = \prod_{i=1}^N p(\bar{V}_i | D_i \mathcal{J}). \quad (13)$$

and

$$p(\bar{V}_i | D_i \mathcal{J}) = \frac{\int_0^{d_i} p(D_i | D_i^* \mathcal{J}) p(D_i^* | \mathcal{J}) dD_i^*}{\int_0^\infty p(D_i | D_i^* \mathcal{J}) p(D_i^* | \mathcal{J}) dD_i^*} \quad (14)$$

As mentioned, the distributions $p(D_i | D_i^* \mathcal{J})$ encode our knowledge about the measurement model. Reasonable choices would be the Gaussian distribution or a Gaussian contaminated by an outlier process. Both of these approaches are evaluated in section 5. Another interesting option would be multi-modal distributions. The prior $p(D_i^* | \mathcal{J})$ encodes some geometric knowledge about the depths in the scene. In all the examples presented a bounding volume was given so we assumed a uniform distribution of D_i^* inside that volume.

If we write $\pi_i = p(\bar{V}_i | D_i \mathcal{J})$ then the evidence for visibility is given by:

$$e(V | D_1 \cdots D_N \mathcal{J}) = \log \frac{1 - \pi_1 \cdots \pi_N}{\pi_1 \cdots \pi_N}. \quad (15)$$

In the following section we point out an interesting connection between the probabilistic visibility approach and one of the classic methods in the Computer Graphics literature for merging range data.

4.5.1 Signed distance functions

In [12], Curless and Levoy compute signed distance functions from each depth-map (positive towards the camera and negative inside the scene) whose weighted average is then stored in a 3d scalar field. So if $w_i(\mathbf{x})$ represents the confidence of depth measurement $D_i(\mathbf{x})$ in the i -th sensor, the 3d scalar field they compute is:

$$F(\mathbf{x}) = \sum_{i=1}^N w_i(\mathbf{x}) (d_i(\mathbf{x}) - D_i(\mathbf{x})) \quad (16)$$

The zero level of $F(\mathbf{x})$ is then computed using marching cubes. While this method provides quite accurate results it has a drawback: For a set of depth maps around a closed object, distances from opposite sides interfere with each other. To avoid this effect [12] actually clamps the distance on either side of a depth map. The distance must be left un-clamped far enough behind the depth map so that all distance functions contribute to the zero-level crossing, but not too far so as to compromise the reconstruction of thin structures. This limitation is due to the fact that the method implicitly assumes that the surface has low relief or that there are no self-occlusions. This can be expressed in several ways but perhaps the most intuitive is that every optic ray from every sensor intersects the surface only once. This means that if a point \mathbf{x} is visible from at least one sensor then it must be visible from all sensors (see figure 12). Using this assumption, an analysis similar to the one in the previous section leads to some a surprising insight into the algorithm. More precisely, if we set the prior probability for visibility to $p(V) = 0.5$ and assume the logistic distribution for sensor noise, *i.e.*

$$p(D_i, D_i^* | \mathcal{I}) \propto \operatorname{sech} \left(\frac{D_i^* - D_i}{2w_i} \right)^2 \quad (17)$$

then the probabilistic evidence for V given all the data exactly corresponds to the right hand side of (16). In other words, the sum of signed distance functions of [12] can be seen as an accumulation of probabilistic evidence for visibility of points in space, given a set of noisy measurements of the depth of the 3d scene. This further reinforces the usefulness of probabilistic evidence for visibility.

4.6 Deformable models

In a similar way to the MRF framework in section 4.1, the deformable model framework [27] allows us to search for an optimal surface S^* that is a minimizer of some user defined energy function E . In general, this energy will be non-convex with possible local optima. In our case, the optimization problem is posed as follows: find the surface S^* of \mathbb{R}^3 that minimizes the energy $E(S)$ defined as:

$$E(S) = E_{ext}(S) + E_{int}(S), \quad (18)$$

where $E_{ext}(S)$ is the external energy term related to the photo-consistency 3d map and $E_{int}(S)$ is the internal energy term or regularization term, *i.e.* a smooth prior on the types of surfaces that we expect. Minimizing Eq. (18) means finding a surface S^* such that satisfies the Euler equation:

$$\nabla E(S^*) = \nabla E_{ext}(S^*) + \nabla E_{int}(S^*) = 0. \quad (19)$$

Equation (19) establishes the equilibrium condition for an optimal solution and can also be seen as a force balance equation:

$$\mathbf{F}_{ext}(S^*) + \mathbf{F}_{int}(S^*) = 0 \quad (20)$$

with $\mathbf{F}_{ext}(S) = \nabla E_{ext}(S)$ and $\mathbf{F}_{int}(S) = \nabla E_{int}(S)$. A solution to Eq. (20) can be found by introducing a time variable t for the surface S and solving the following differential equation:

$$\frac{\partial S}{\partial t} = \mathbf{F}_{ext}(S) + \mathbf{F}_{int}(S). \quad (21)$$

The discrete version becomes:

$$S^{k+1} = S^k + \Delta t(\mathbf{F}_{ext}(S^k) + \mathbf{F}_{int}(S^k)). \quad (22)$$

Once we have sketched the energies that will drive the process, we need to make a choice for the representation of the surface S . This representation defines the way the deformation of the surface is performed at each iteration. We choose the triangular mesh because of its simplicity and well known properties, but other options such as implicit surface representations can be used [25].

To completely define the deformation framework, we need an initial value of S , *i.e.* an initial surface S^0 that will evolve under the different forces until convergence. S^0 can range from the most basic initial shape such as a bounding box, to a better one like the visual hull, or an even better one such as the provided by the MRF framework in section 4.1.

In the following we describe how to derive the external force from the photo-consistency 3d map and the internal force on a triangular mesh.

4.6.1 External force: octree-based gradient vector flow

The external force is directly linked to the photo-consistency 3d map previously described in section 3. We want this force to drive the surface to high photo-consistency locations. However the volume of photo-consistency $\mathcal{C}(\mathbf{x})$ itself cannot be used as a force to drive the deformable model. A typical force would be the gradient of $\mathcal{C}(\mathbf{x})$, *i.e.* $\mathbf{F}_{ext}(\mathbf{x}) = \nabla \mathcal{C}(\mathbf{x})$. The main objection is that it is a very local force defined only in the vicinity of the object surface. A better solution is to use a gra-

dient vector flow (GVF) field derived from the photo-consistency in order drive the deformable model.

The GVF field was introduced by [52] as a way to overcome a difficult problem of traditional deformable models: the capture range of the data term. This problem is caused by the local definition of the force, and the absence of an information propagation mechanism. To eliminate this drawback, and for all the forces derived from the gradient of a scalar field, they proposed to generate a vector field force that propagates the gradient information. The GVF of a scalar field $f(x, y, z) : \mathbb{R}^3 \mapsto \mathbb{R}$ is defined as the vector field $\mathbf{F} = [u(x, y, z), v(x, y, z), w(x, y, z)] : \mathbb{R}^3 \mapsto \mathbb{R}^3$ that minimizes the following energy functional E_{GVF} :

$$E_{GVF} = \int \mu \|\nabla \mathbf{F}\|^2 + \|\mathbf{F} - \nabla f\|^2 \|\nabla f\|^2, \quad (23)$$

where μ is the weight of the regularization term and $\nabla \mathbf{F} = [\nabla u, \nabla v, \nabla w]$. The solution to this minimization problem has to satisfy the Euler equation:

$$\mu \nabla^2 \mathbf{F} - (\mathbf{F} - \nabla f) \|\nabla f\|^2 = 0, \quad (24)$$

where $\nabla^2 \mathbf{F} = [\nabla^2 u, \nabla^2 v, \nabla^2 w]$ and ∇^2 is the Laplacian operator. A numerical solution can be found by introducing a time variable t and solving the following differential equation:

$$\frac{\partial \mathbf{F}}{\partial t} = \mu \nabla^2 \mathbf{F} - (\mathbf{F} - \nabla f) \|\nabla f\|^2. \quad (25)$$

The GVF can be seen as the original gradient smoothed by the action of a Laplacian operator. This smoothing action allows eliminating strong variations of the gradient and, at the same time, propagating it. The degree of smoothing/propagation is controlled by μ . If μ is zero, the GVF will be the original gradient, if μ is very large, the GVF will be a constant field whose components are the mean of the gradient components. Applied to the deformable model problem, the external force \mathbf{F}_{ext} is then found as the solution of the following differential equation:

$$\frac{\partial \mathbf{F}_{ext}}{\partial t} = \mu \nabla^2 \mathbf{F}_{ext} - (\mathbf{F}_{ext} - \nabla C) \|\nabla C\|^2, \quad (26)$$

with μ always fixed to 0.1.

4.6.2 Mesh Control

The goal of the internal force is to regularize the effect of the external forces. Following the formulation by [10], we define the internal energy E_{int} of a surface S as the sum of two terms penalizing for changes in the first and second order derivatives of the surface. A local minimum of the energy $E_{int}(S)$ satisfies the associated Euler-Lagrange equation, which gives us the following form for the internal force:

$$\mathbf{F}_{int}(S) = \gamma_1 \Delta S + \gamma_2 \Delta^2 S, \quad (27)$$

where Δ is the Laplacian operator and Δ^2 is the biharmonic operator. The discrete version of the Laplacian operator $\tilde{\Delta}$ on a triangle mesh can be easily implemented using the umbrella operator, *i.e.* the operator that tries to move a given vertex \mathbf{v} of the mesh to the center of gravity of its 1-ring neighborhood $\mathcal{N}_1(\mathbf{v})$:

$$\tilde{\Delta}\mathbf{v} = \left(\sum_{i \in \mathcal{N}_1(\mathbf{v})} \frac{\mathbf{v}_i}{m} \right) - \mathbf{v}, \quad (28)$$

where \mathbf{v}_i are the neighbors of \mathbf{v} and m is the total number of these neighbors (valence). Concerning the discrete version of the biharmonic operator $\tilde{\Delta}^2$, its derivation is less trivial:

$$\tilde{\Delta}^2\mathbf{v} = \frac{1}{1 + \sum_{i \in \mathcal{N}_1(\mathbf{v})} \frac{1}{mm_i}} \tilde{\Delta}(\tilde{\Delta}\mathbf{v}), \quad (29)$$

The total internal force on a mesh vertex \mathbf{v} is defined as:

$$\mathbf{F}_{int}(\mathbf{v}) = \gamma_1 \tilde{\Delta}\mathbf{v} + \gamma_2 \tilde{\Delta}^2\mathbf{v}. \quad (30)$$

Since the texture force \mathbf{F}_{ext} can sometimes be orthogonal to the surface of the snake, we do not use the force \mathbf{F}_{ext} itself but its projection \mathbf{F}_{ext}^N onto the surface normal \mathbf{n} :

$$\mathbf{F}_{ext}^N(\mathbf{v}) = (\mathbf{n}^\top \cdot \mathbf{F}_{ext}(\mathbf{v}))\mathbf{n}. \quad (31)$$

This avoids problems of coherence in the force of neighbor points and helps the internal force to keep a well-shaped surface.

The evolution process (Eq. 22) at the k^{th} iteration can then be written as the evolution of all the points of the mesh \mathbf{v}_i :

$$\mathbf{v}_i^{k+1} = \mathbf{v}_i^k + \Delta t (\mathbf{F}_{ext}^N(\mathbf{v}_i^k) + \mathbf{F}_{int}(\mathbf{v}_i^k)), \quad (32)$$

where Δt is the time step and α is the weight of the regularization term relative to the external term. Equation (32) is iterated until convergence of all the points of the mesh is achieved. The time step Δt has to be chosen as a compromise between the stability of the process and the convergence time. An additional step of remeshing is done at the end of each iteration in order to maintain a minimum and a maximum distance between neighbor points of the mesh. This is obtained by a controlled decimation and refinement of the mesh. The decimation is based on the edge collapse operator and the refinement is based on the $\sqrt{3}$ -subdivision scheme [28].

5 Experiments

5.1 Depth map evaluation

In order to solve the depth-map computation algorithm described in section 3, we use the TRW-S implementation of Kolmogorov [30]. The proposed implementation, running on a 3.0 GHz machine with an nVidia Quadro graphics card, can evaluate 900 NCC depth slices in 20 seconds for the temple sequence (image resolution 640×480). The TRW-S optimization has a typical run time of 20 seconds for the same images.

For all the experiments we used the following parameter values: $\beta = 1$, $\lambda = 1$, $\phi_U = 0.04$ and $\psi_U = 0.002$. We used an NCC window size of 5×5 .

Fig. 13 illustrates the improvement of the method described in section 3.2 over the voting schemes of [20, 47]. Fig. 13 (b) shows the depth that would be determined by simply taking the NCC peak with the greatest score. The new method, implemented here with $K = 9$ peaks, is able to select the peak corresponding to true surface peak from the ranked candidate peaks and Fig. 13 (d) illustrates that a significant proportion of the true surface peaks are not the absolute maximum. We also observe that pixels are correctly labeled with the *unknown* state along occlusion boundaries and along areas such as the back wall of the temple and edges of the pillars where the surface normal is oriented away from the camera. Looking at the rendering of this depth-map and its neighbor, Fig. 13(e-g), we can observe that very few erroneous depths are recovered and we observe that the combination of the two depths maps align and complement each other rather than attempting to fill in the holes on the individual depth-maps which would impact the subsequent multi-view stereo global optimization.

Fig. 14 shows the results on the ‘cones’ dataset which forms part of the standard dense stereo evaluations images and consists of a single stereo pair with the left image shown. The depth-map again shows a high degree of detail on textured surfaces and we correctly identify occlusion boundaries with the *unknown* state. Further more the algorithm also correctly textures the failure modes of NCC by returning the *unknown* state in texture-less regions where the matching fails to accurately localize the surface.

5.2 Multi-view stereo evaluation

In order to evaluate the improvement of the depth-map estimation algorithm of section 3.2 for multi-view stereo we ran the algorithm on the standard evaluation ‘temple’ dataset. The following table provides the accuracy and completeness measures of [40] against the ground-truth data for the object. In terms of both accuracy and completeness the results provide a significant improvement in both the sparse ring and ring datasets. In particular we observe that the results for the sparse ring of-

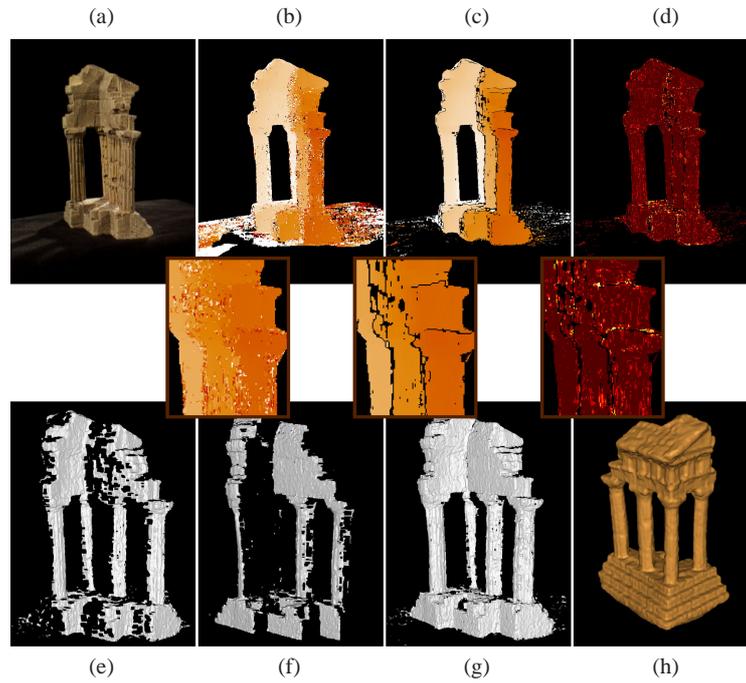


Fig. 13 Results of the depth map estimation algorithm. Two neighboring images are combined with the reference image (a). If we simply took the NCC peak with the maximum score, as in [20], we would obtain (b). The result of the algorithm used in section 3.2 (c) shows a significant reduction in noise. We have corrected noisy estimates of the surface and the *unknown* state has also been used to clearly denote occlusion boundaries and remove poorly matched regions. The number of the correct surface peak returned, ranked by NCC score, is displayed in (d) where dark red indicates the peak with the greatest score. The rendered depth-map is shown in (e) along with the neighboring depth-map (f) with (g) showing the two superimposed. The final reconstruction (h) for the sparse temple sequence (16 images) of [40]

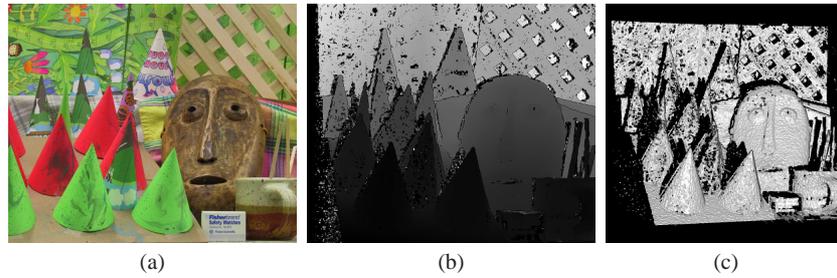


Fig. 14 Single view stereo results for the ‘Cones’ data set. The left image of the stereo pair is shown in (a) with the recovered depth-map in (b), rendered in (c)

fer greater accuracy than the other algorithms [40] running on the ring sequence (3 times as many images) with the exception of [20].

	Accuracy / Completeness		
	Full (312 images)	Ring (47 images)	SparseRing (16 images)
proposed method	0.41mm / 99.9%	0.48mm / 99.4%	0.53mm / 98.6%

5.3 Digitizing works of art

The proposed pipeline has been used to reconstruct a bronze statue located in the British museum in London from holiday photographs. The photographs were taken by a hand held camera during normal visiting hours (see Fig. 15). This led to the statue being photographed with cluttered and changing background. The camera motion was automatically recovered using a structure-from-motion technique [55]. The bottom row of figure 15 shows the intermediate results obtained while reconstructing the statue. From left to right, we show a rendering of the 3d map of photo-consistency (section 3), the initial surface obtained using graph-cuts (section 4.1), the refined surface obtained with the deformable model (section 4.6), and the same surface textured mapped from the input photographs using [20]. Note how, even with a noisy photo-consistency 3d volume, the graph-cut solution is able to extract a very detailed surface. However, this surface has discretization artifacts due to the binary nature of the graph-cut solution. These artifacts are completely removed when the surface is refined using a deformable model. A similar refinement step is also used in [17].

We present a second sequence of 72 images of a “*crouching man*” sculpture made of plaster by the modern sculptor Antony Gormley (see Fig. 16 top).

The image resolution is 5 Mpix and the camera motion was recovered by standard structure from motion techniques [55] and further refined using a silhouette-based technique [21]. The object exhibits significant self-occlusions, a large concavity in the chest and two thin legs which make it a very challenging test to validate the new ballooning term. The first step in the reconstruction process is to compute a set of depth-maps from the input images. This process is by far the most expensive of the whole pipeline in terms of computation time. A single depth-map takes between 90 and 120 seconds, the overall computation time being close to 2 hours. Once the depth-maps are computed, a 3d octree grid can be built (see Fig. 9 left) together with the discontinuity cost and the labeling cost (see Fig. 9 middle and right respectively). Because of the octree grid, we are able to use up to 10 levels of resolution to compute the graph-cut, *i.e.* the equivalent of a regular grid of 1024^3 voxels. We show in figure 16 some of the images used in the reconstruction (top), the result using an implementation of [48] (middle) and the reconstruction result of the proposed method (bottom). We can appreciate how the constant ballooning term introduced in [48] is unable to reconstruct correctly the feet and the concavities at the same time. In order to recover thin structures such as the feet, the ballooning term needs to be

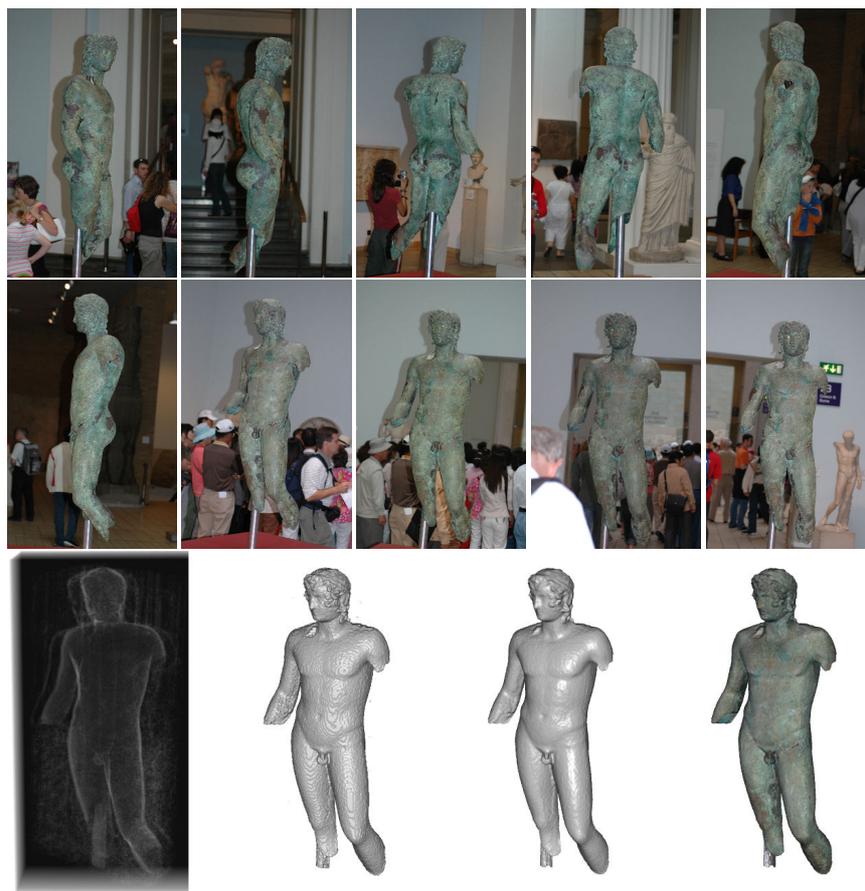


Fig. 15 Statue of a young man, Mimaut Collection. Bronze, Roman copy of the 1st century BC after a Greek original. From Ziphteh, near Tell Atrib (ancient Athribis), Egypt. The sequence was acquired with a hand held camera in the British museum with no special requirements. Background is extremely cluttered. The object of interest is both in the center of the photographs and in focus. Top and middle rows show a few samples of the original sequence. Last row shows from left to right, 3d map of photo-consistency described in section 3, surface extracted using graph-cuts (section 4.1), surface refined using a deformable model (section 4.6) and surface textured-map from the original photographs using [20].

stronger. But even before the feet are fully recovered, the concavities start to over inflate.

Finally we show in figure 17 the effect of having an outlier component in the noise model of the depth sensor when computing the volume of evidence of visibility. The absence of an outlier model that is able to cope with noisy depth estimates appears in the volume of visibility as tunnels “drilled” by the outliers (see Fig. 17 center). Adding an outlier term clearly reduces the tunneling effect while preserving the concavities (see Fig. 17 right).

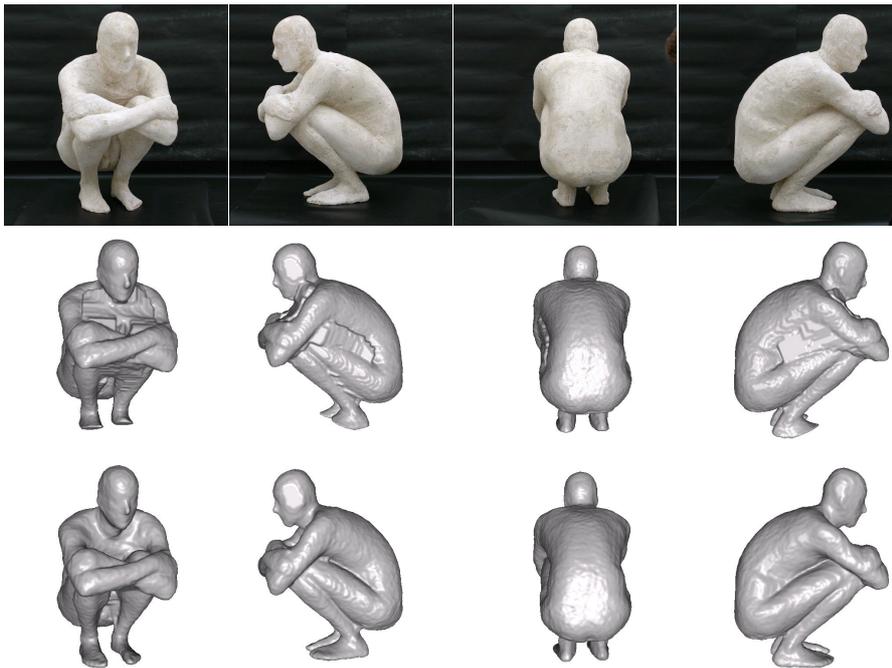


Fig. 16 Comparison of the improvement obtained with the visibility-driven ballooning term. Plaster model of a crouching man by Antony Gormley, 2006. Top: some of the input images. Middle: views of reconstructed model using the technique of [48] with a constant ballooning term. No constant ballooning factor is able to reconstruct correctly the feet and the concavities at the same time. Bottom: views of reconstructed model using the intelligent ballooning proposed by [22] and shown in Fig.17 right.

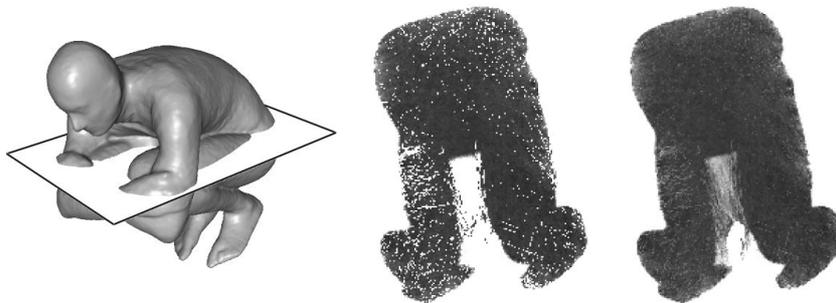


Fig. 17 Comparison of two different inlier/outlier ratios for the depth sensor noise model. Left: 3d location of one slice of the volume of “evidence of visibility”. Middle: the sensor model is a pure Gaussian without any outlier model. Outliers “drill” tunnels in the visibility volume. Right: the sensor model takes into account an outlier model. The visibility volume is more robust against outliers while the concavities are still distinguishable.

6 Discussion

We have described a formulation to multi-view stereo that splits the problem into a well defined pipeline of 3 building blocks: camera calibration, computation of a 3d volume of photo-consistency and extraction of a surface from the photo-consistency volume. In this chapter we have particularly focus on how to compute a 3d volume of photo-consistency, and how to extract a 3d surface from the photo-consistency volume. The main advantages of such an approach are its simplicity and room for improvement, since it uses two very standard off-the-shelf algorithms such as dense stereo and 3d segmentation algorithms. The main disadvantage is the rather simplistic photo-consistency metric, which leads to poor performance in challenging conditions such as sparse set of photographs or poorly textured surfaces. These problems are partially mitigated by explicitly accounting for the failure modes of the window matching technique in section 3. However, a more thorough matching technique using a local planarity assumption such as [17] would also greatly improve results in challenging scenes. The framework we describe in this chapter has been widely adopted by a variety of multi-view stereo algorithms [7, 8, 18, 20, 24, 29, 38, 42, 47]. This can be mainly justified by the simplicity of the approach, but also by the flexibility that it offers, *e.g.* when trying to optimally fuse the photo-consistency cue with apparent contours as proposed in [29].

Appendix. Interpretation of signed distance functions.

Using the predicates we have already defined, the assumption of no self-occlusion can be expressed by

$$V \leftrightarrow \forall i V_i. \quad (33)$$

From (10) and (33) we see that if a point \mathbf{x} is visible (invisible) from one sensor it is visible (invisible) from all sensors, i.e. $V_1 \leftrightarrow \dots \leftrightarrow V_N \leftrightarrow V$. Let \mathcal{I} stand for the prior knowledge which includes the geometric description of the problem and (33). Given (33) events $D_1 \dots D_N$ are independent under the knowledge of V or \bar{V} which means that using Bayes' theorem we can write:

$$p(V | D_1 \dots D_N \mathcal{I}) = \frac{p(V | \mathcal{I}) \prod_{i=1}^N p(D_i | V \mathcal{I})}{p(D_1 \dots D_N | \mathcal{I})} \quad (34)$$

Obtaining the equivalent equation for \bar{V} and dividing with equation (34) and taking logs gives us:

$$e(V | D_1 \dots D_N \mathcal{I}) = e(V | \mathcal{I}) + \sum_{i=1}^N \log \frac{p(D_i | V \mathcal{I})}{p(D_i | \bar{V} \mathcal{I})}. \quad (35)$$

By several applications of Bayes' theorem we get:

$$e(V | D_1 \cdots D_N \mathcal{I}) = \sum_{i=1}^N \log \frac{\alpha_i}{\beta_i} - (N-1)e(V | \mathcal{I}). \quad (36)$$

where $\alpha_i = \int_{d_i}^{\infty} p(D_i, D_i^* | \mathcal{I}) dD_i^*$ and $\beta_i = \int_0^{d_i} p(D_i, D_i^* | \mathcal{I}) dD_i^*$. We now set $e(V | \mathcal{I}) = 0$ and assume the noise model is given by the logistic function

$$p(D_i, D_i^* | \mathcal{I}) \propto \operatorname{sech} \left(\frac{D_i^* - D_i}{2w_i} \right)^2. \quad (37)$$

Using standard calculus one can obtain the following expression for the evidence

$$e(V | D_1 \cdots D_N \mathcal{I}) = \sum_{i=1}^N w_i (d_i - D_i), \quad (38)$$

equal to the average of the distance functions used in [12].

References

1. Baumgart, B.G.: Geometric modelling for computer vision. Ph.D. thesis, Stanford University (1974)
2. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive GMMRF model. In: Proc. 8th Europ. Conf. on Computer Vision (ECCV), pp. 428–441 (2004)
3. Boissonnat, J.D., Faugeras, O., Lebras, E.: Representing stereo data with the delaunay triangulation. *Artificial Intelligence* **44**, 41–87 (1990)
4. Bolitho, M., Kazhdan, M., Burns, R., Hoppe, H.: Multilevel streaming for out-of-core surface reconstruction. In: Eurographics Symposium on Geometry Processing, pp. 69–78 (2007)
5. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Proc. 9th Intl. Conf. on Computer Vision (ICCV), pp. 26–33 (2003)
6. Boykov, Y., Lempitsky, V.: From photohulls to photoflux optimization. In: Proc. British Machine Vision Conference, to appear, pp. 1149–1158 (2006)
7. Bradley, D., Boubekur, T., Heidrich, W.: Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2008)
8. Campbell, N., Vogiatzis, G., Hernández, C., Cipolla, R.: Using multiple hypotheses to improve depth-maps for multi-view stereo. In: Proc. 10th Europ. Conf. on Computer Vision (ECCV) (2008)
9. Cohen, L.D.: On active contour models and balloons. *CVGIP: Image Understanding* **53**(2), 211–218 (1991)
10. Cohen, L.D., Cohen, I.: Finite element methods for active contour models and balloons for 2-D and 3-D images. *PAMI* **15**(11), 1131–1147 (1993)
11. Cornelis, N., Leibe, B., Cornelis, K., Gool, L.: 3d urban scene modeling integrating recognition and reconstruction. *Intl. Journal of Computer Vision* **2-3**(78), 121–141 (2008)
12. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. *SIGGRAPH* pp. 303–312 (1996)
13. Faugeras, O., Keriven, R.: Variational principles, surface evolution, pdes, level set methods and the stereo problem. *IEEE Transactions on Image Processing* **7**(3), 335–344 (1998)

14. Favaro, P., Soatto, S.: 3-D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur. Springer (2006)
15. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Reconstructing building interiors from images. In: Proc. IEEE Conf. ICCV (2009)
16. Furukawa, Y., Ponce, J.: Carved visual hulls for image-based modeling. In: Proc. 9th Europ. Conf. on Computer Vision (ECCV), vol. 1, pp. 564–577 (2006)
17. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2007)
18. Goesele, M., Curless, B., Seitz, S.: Multi-view stereo revisited. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 2402–2409 (2006)
19. Habbeke, M., Kobbelt, L.: A surface-growing approach to multi-view stereo reconstruction. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2007)
20. Hernández, C., Schmitt, F.: Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding* **96**(3), 367–392 (2004)
21. Hernández, C., Schmitt, F., Cipolla, R.: Silhouette coherence for camera calibration under circular motion. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(2), 343–349 (2007)
22. Hernández, C., Vogiatzis, G., Cipolla, R.: Probabilistic visibility for multi-view stereo. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2007)
23. Hernández, C., Vogiatzis, G., Cipolla, R.: Multi-view photometric stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(1), 548–554 (2008)
24. Hornung, A., Kobbelt, L.: Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 503–510 (2006)
25. Ilic, S., Fua, P.: Implicit meshes for surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(2), 328–333 (2006)
26. Jaynes, E.: *Probability Theory, The Logic of Science*. Cambridge University Press (2003)
27. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* **1**, 321–332 (1988)
28. Kobbelt, L.: $\sqrt{3}$ -subdivision. In: SIGGRAPH 2000, pp. 103–112 (2000)
29. Kolev, K., Cremers, D.: Integration of multiview stereo and silhouettes via convex functionals on convex domains. In: Proc. 10th Europ. Conf. on Computer Vision (ECCV), pp. 752–765 (2008)
30. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(10), 1568–1583 (2006). DOI <http://dx.doi.org/10.1109/TPAMI.2006.200>
31. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(2), 147–159 (2004)
32. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. *Intl. Journal of Computer Vision* **38**(3), 199–218 (2000)
33. Lempitsky, V., Boykov, Y., Ivanov, D.: Oriented visibility for multiview reconstruction. In: Proc. 9th Europ. Conf. on Computer Vision (ECCV), vol. 3, pp. 226–238 (2006)
34. Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., Fulk, D.: The digital michelangelo project: 3d scanning of large statues. In: Proc. of the ACM SIGGRAPH, p. 1522 (2000)
35. Lhuillier, M., Quan, L.: A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(3), 418–433 (2005)
36. M., G., N., S., B., C., H., H., S., S.: Multi-view stereo for community photo collections. In: Proceedings of ICCV 2007 (2007)
37. Pollefeys, M., Gool, L.J.V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. *International Journal of Computer Vision* **59**(3), 207–232 (2004)
38. Pollefeys, M., Nistér, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim S. J. and Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3d reconstruction from video. *Intl. Journal of Computer Vision* **78**(2-3), 143–167 (2008)

39. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *IJCV (The International Journal of Computer Vision)* **72**(2), 179–193 (2007)
40. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 519–528 (2006)
41. Sinha, S., Pollefeys, M.: Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In: *Proc. 10th Intl. Conf. on Computer Vision (ICCV)*, vol. 1, pp. 349–356 (2005)
42. S.N., S., P., M., M., P.: Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In: *Proc. 11th Intl. Conf. on Computer Vision (ICCV)* (2007)
43. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring image collections in 3d. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)* (2006)
44. Steger, E., Kutulakos, K.N.: A theory of refractive and specular 3d shape by light-path triangulation. *International Journal of Computer Vision* **76**(1) (2008)
45. Strecha, C., von Hansen, W., Gool, L.V., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2008)
46. Tran, S., Davis, L.: 3d surface reconstruction using graph cuts with surface constraints. In: *Proc. 9th Europ. Conf. on Computer Vision (ECCV)*, vol. 2, pp. 218–231 (2006)
47. Vogiatzis, G., Hernández, C., Torr, P., Cipolla, R.: Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(12), 2241–2246 (2007)
48. Vogiatzis, G., Torr, P., Cipolla, R.: Multi-view stereo via volumetric graph-cuts. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 391–398 (2005)
49. Weise, T., Leibe, B., Gool, L.V.: Fast 3d scanning with automatic motion compensation. In: *CVPR '07: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2007)
50. Witkin, A.: Recovering surface shape and orientation from texture. *Artificial Intelligence* **17**(1-3), 17–45 (1981)
51. Woodham, R.: Photometric method for determining surface orientation from multiple images. *Optical Engineering* **19**(1), 139–144 (1980)
52. Xu, C., Prince, J.L.: Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing* pp. 359–369 (1998)
53. Zebedin, L., Bauer, J., Karner, K., Bischof, H.: Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 873–886 (2008)
54. Zhang, L., Snavely, N., Curless, B., Seitz, S.M.: Spacetime faces: High-resolution capture for modeling and animation. *ACM Annual Conference on Computer Graphics* pp. 548–558 (2004)
55. Zisserman, A., Hartley, R.: *Multiple View Geometry*. Springer-Verlag (2000)