ADVERTIMENT. La consulta d’aquesta tesi queda condicionada a l’acceptació de les següents condicions d’ús: La difusió d’aquesta tesi per mitjà del servei TDX (www.tesisenxarxa.net) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats enmarcats en activitats d’investigació i docència. No s’autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d’un lloc aliè al servei TDX. No s’autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA. La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDX (www.tesisenred.net) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDX. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDX (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING. On having consulted this thesis you’re accepting the following use conditions: Spreading this thesis by the TDX (www.tesisenxarxa.net) service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the TDX service. Introducing its content in a window or frame foreign to the TDX service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it’s obliged to indicate the name of the author.
Ph.D. Dissertation

Surface Reconstruction for Multi-View Video

Jordi Salvador Marcos

Thesis supervisor:
Dr. Josep R. Casas Pla

Department of Signal Theory and Communications
Universitat Politècnica de Catalunya

Barcelona, July 2011
Chapter 9

Surface Interpolation Results

The algorithm for linear interpolation of a surface presented in Chapter 8 has been designed in order to provide a topologically correct triangular mesh that connects an input set of oriented points. The natural idea is to use this method for oriented points extracted with any of the surface sampling techniques proposed in the first part of this dissertation. However, in order to characterize its performance, in this chapter we present an evaluation based on a comparison of several surface reconstructions with range data, which have been obtained from [The Stanford 3D Scanning Repository, 2010].

First, the different techniques used for comparison are listed, which also provide a triangular mesh from sets of oriented 3D points. Next, the datasets and the metrics used for comparing them to our proposed technique are presented and quantitative results are provided for the six used datasets, for which a reference surface represented as a triangular mesh is available and used as ground-truth. Finally, a qualitative evaluation of the quality of the surfaces obtained with all methods for each used dataset is also introduced.

9.1 Methods

In this section, we present the techniques for triangulation of a surface from a set of oriented points that we have used in the evaluation of the technique presented in Chapter 8. The goal is to compare the results obtained with several approaches falling in the two categories presented in the Chapter 2 (Section 2.3) and in more detail in Section 8.1.2 in the previous chapter: propagation-based and marching cubes-based methods. All the following algorithms, except for the proposed method and ReOP, are implemented in the MeshLab software [Cignoni et al., 2008].
9.1.1 Propagation methods

From this category, the ball pivoting algorithm, restricted and oriented propagation and the proposed method have been used.

**BPA.** The ball pivoting algorithm [Bernardini et al., 1999] is related to the so-called alpha-shapes, which can be seen as a generalization of a convex hull [Edelsbrunner and Mücke, 1994]. In this method, a virtual ball of a user-specified radius scans a cloud of oriented points and obtains triangular faces connecting them in a surface propagation approach.

In our experiments, the initial value for the ball radius is automatically adjusted, and three iterations are then applied with increasing radius in steps of 0.5% of the bounding box diagonal.

**ReOP.** Restricted and oriented propagation [Suau et al., 2010] is a fast method based on voxel propagation, with the main drawback of a non-manifold output mesh, which renders this approach useful when targeting visualization, but not when the reconstructed surface has to be further processed.

In our experiments, only the parameter adjusting the voxel size is adjusted, leaving the 6-OCT pattern constant throughout all the executions. The rule for choosing the voxel size is such that at least 96% of the points are not sharing a voxel with other points.

**Proposed.** The proposed method presented in Chapter 8 creates an initial triangular face and iteratively propagates its edges to generate new faces under a set of rules for topological correctness.

The only parameter that has been adjusted is the number of results $k$ for nearest-neighbor search, setting it to a value such that no holes appear in the resulting surfaces (ranging from 5 to 20 in our experiments).

9.1.2 Marching cubes-based methods

From this category, RIMLS-MC and Poisson reconstruction have been used, two methods that have in common the use of marching cubes [Lorensen and Cline, 1987] for extracting the final surface.

**RIMLS-MC.** Robust Implicit Moving Least Squares-Marching Cubes [Oztireli et al., 2009] obtains a Moving Least Squares surface defined by the point set as
a robust implicit extension of Moving Least Squares that preserves sharp features, by using non-linear regression. The surface is finally extracted with marching cubes.

In our experiments, the scale of the spatial low-pass filter, relative to the radius—local point spacing—of the vertices is set to 4. The grid resolution, as in ReOP, is manually adjusted until the number of vertices at the output is practically equal to that at the input. It has also been visually checked that the presence of holes due to volumetric under-sampling remains under reasonable limits.

**Poisson-Rec.** *Poisson reconstruction* [Kazhdan et al., 2006] solves a Poisson equation for volumetric occupancy trying to best approximate the vector field defined by the samples. Then, it obtains a smooth surface after applying an octree-based implementation of marching cubes to the resulting occupancy function.

In our experiments, only one parameter has been tuned. This is the depth of the octree used for extracting the final surface. Again, the criterion has been that the number of output vertices was approximately equal to the number of surface samples at the input.

### 9.2 Datasets and Metrics

In this section, we first present the six datasets used for both the quantitative and the qualitative comparisons of the polygon meshes obtained with each of the techniques listed in the previous section. Then, the metrics used for comparing the performance of each method are also presented.

#### 9.2.1 Datasets

The experiments presented below are carried out with six datasets containing surfaces represented as oriented points, most of them provided by the Stanford 3D Scanning Repository [The Stanford 3D Scanning Repository, 2010]. These are listed in Table 9.1.

For all these datasets, a reference meshing is provided along with the oriented points, which is used as ground-truth for the quantitative comparison below. These datasets include both irregular and uniform distributions of surface points. Thus, they provide a sufficient measure of the performance of the proposed method under different scenarios.
### Table 9.1: Datasets used for quantitative evaluation of our proposed surface interpolation algorithm

<table>
<thead>
<tr>
<th>Name</th>
<th># Points</th>
<th>Points distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>drill</td>
<td>1,960</td>
<td>Uniform</td>
</tr>
<tr>
<td>bunny</td>
<td>34,834</td>
<td>Uniform</td>
</tr>
<tr>
<td>armadillo</td>
<td>172,974</td>
<td>Uniform</td>
</tr>
<tr>
<td>hand</td>
<td>327,323</td>
<td>Irregular</td>
</tr>
<tr>
<td>dragon</td>
<td>437,645</td>
<td>Very irregular</td>
</tr>
<tr>
<td>happy</td>
<td>543,652</td>
<td>Very irregular</td>
</tr>
</tbody>
</table>

#### 9.2.2 Metrics

The RMS Hausdorff distance metric is used to evaluate the similarity between the obtained surface and a ground-truth surface, giving an idea of the accuracy of the reconstruction. More details about this metric can be found in Appendix C.

A second parameter taken into account is the overall computation time, which does not include the time required to read the oriented points from the input file nor the time required to write the polygon mesh to the output file.

The third and last figure that has been taken for comparison is the memory footprint of each method. The memory measurements are only valid as an approximation to the actual values, since they have been obtained as a multiple of one thousandth part of the total amount of available memory. However, they reflect the trends with respect to memory efficiency of each compared method.

Thus, the proposed surface interpolation algorithm is quantitatively compared to the methods introduced above using these three figures. For reference, all the experiments are executed on a 64-bit Intel Xeon CPU 3.0 GHz with 32 GB RAM.

For the qualitative comparison, visual inspection of both the overall mesh and some details in challenging areas are considered in order to evaluate the performance of each method.

#### 9.3 Results

In this section, we present the quantitative and qualitative results obtained with the proposed technique and each of the remaining five methods used for comparison. Details of the datasets presented above, which have also been used for the qualitative comparison, are introduced in the qualitative comparison.
9.3 Results

<table>
<thead>
<tr>
<th>Method</th>
<th>drill</th>
<th>bunny</th>
<th>armadillo</th>
<th>hand</th>
<th>dragon</th>
<th>happy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time</strong></td>
<td>1.66</td>
<td>36.06</td>
<td>170.42</td>
<td>211.92</td>
<td>180.32</td>
<td>215.24</td>
</tr>
<tr>
<td>RIMLS-MC</td>
<td>-</td>
<td>-</td>
<td>96</td>
<td>192</td>
<td>256</td>
<td>288</td>
</tr>
<tr>
<td>Mem.</td>
<td>1747</td>
<td>526</td>
<td>249</td>
<td>234</td>
<td>279</td>
<td>344</td>
</tr>
<tr>
<td><strong>RMS</strong></td>
<td>0.66</td>
<td>13.00</td>
<td>35.64</td>
<td>38.81</td>
<td>43.61</td>
<td>49.55</td>
</tr>
<tr>
<td>Poisson-Rec</td>
<td>-</td>
<td>-</td>
<td>224</td>
<td>480</td>
<td>640</td>
<td>704</td>
</tr>
<tr>
<td>Mem.</td>
<td>2532</td>
<td>1367</td>
<td>591</td>
<td>543</td>
<td>874</td>
<td>1232</td>
</tr>
<tr>
<td><strong>RMS</strong></td>
<td>0.15</td>
<td>48.62</td>
<td>347.00</td>
<td>935.36</td>
<td>1846.49</td>
<td>2988.64</td>
</tr>
<tr>
<td>BPA</td>
<td>-</td>
<td>-</td>
<td>64</td>
<td>128</td>
<td>192</td>
<td>224</td>
</tr>
<tr>
<td>Mem.</td>
<td>506</td>
<td>99</td>
<td>61</td>
<td>92</td>
<td>105</td>
<td>144</td>
</tr>
<tr>
<td><strong>RMS</strong></td>
<td>0.70</td>
<td>0.81</td>
<td>4.10</td>
<td>8.73</td>
<td>14.36</td>
<td>14.89</td>
</tr>
<tr>
<td>ReOP</td>
<td>-</td>
<td>-</td>
<td>288</td>
<td>1088</td>
<td>1696</td>
<td>1920</td>
</tr>
<tr>
<td>Mem.</td>
<td>464</td>
<td>94</td>
<td>61</td>
<td>163</td>
<td>56</td>
<td>116</td>
</tr>
<tr>
<td><strong>RMS</strong></td>
<td>0.03</td>
<td>0.92</td>
<td>3.06</td>
<td>8.12</td>
<td>10.11</td>
<td>15.44</td>
</tr>
<tr>
<td>Proposed</td>
<td>-</td>
<td>-</td>
<td>96</td>
<td>192</td>
<td>288</td>
<td>320</td>
</tr>
<tr>
<td><strong>RMS</strong></td>
<td>410</td>
<td>90</td>
<td>40</td>
<td>43</td>
<td>103</td>
<td>113</td>
</tr>
</tbody>
</table>

Table 9.2: Computation time, memory usage and accuracy for meshing each of the six sets of oriented points. The execution time is shown in seconds and the memory usage in megabytes, whereas the RMS Hausdorff distance is shown in units of a millionth part of the main diagonal of the bounding box of each dataset. In bold text, the shortest execution time, the least memory usage and the best accuracy for each dataset.

9.3.1 Quantitative comparison

In Table 9.2, the execution time, approximate memory usage and RMS Hausdorff distance—in all cases, the smaller, the better—are shown for each of the datasets and the techniques used for comparison. About the units, execution time is measured in seconds and memory usage in megabytes, whereas the RMS Hausdorff distance is measured in units of a millionth part \((10^{-6})\) of the diagonal of the corresponding bounding box. The shortest computation time, smallest memory footprint and best accuracy are written in bold font for each dataset. Due to the small resolution of the memory usage measurements, the memory footprint of the datasets drill and bunny could not be obtained.

As shown in Table 9.2, the proposed method is always showing either the best computation time or the best accuracy among the tested methods. In terms of computation time, the properly created \(kd\)-tree proves as a suitable data structure for spatial queries, when compared to other propagation-based schemes. In terms of quality, when compared to marching cubes-based methods, the utilization of the exact position of surface samples makes propagation-based techniques much more accurate, whereas among these latter techniques, the proposed one provides in most cases the surface closest to the reference one due to the topologically correct
configuration of the resulting triangular faces.

With respect to the memory footprint, the comparison might be biased: the methods executed from Meshlab –BPA, RIMLS-MC and Poisson-Rec– have been measured by subtracting the initial memory usage of the program, before loading the datasets and starting the meshing algorithm. It might be possible that some memory is released by the program during the execution of potentially memory-demanding methods, or that some data structures actually used by the meshing methods are pre-loaded when the program starts. Thus, we cannot take solid conclusions about the memory usage beyond that, in general, methods requiring a global voxelization of the working volume –Poisson-Rec and ReOP– are more memory demanding. In general, the propose method offers a good computation time vs. memory footprint trade-off.

To sum up, the proposed method is the best in terms of accuracy and execution time, in general, being closely followed by ReOP in most cases and superseded by this method in the few remaining ones. In terms of memory usage, it was not possible to offer an accurate comparison with all of the methods, but it compares favorably to ReOP and Poisson-Rec, the closest contenders in terms of computation time.

### 9.3.2 Qualitative comparison

For the following experiments, a short introduction to each dataset is presented. Then, some qualitative valuations are drawn from the results depicted in the corresponding figures, presenting both general views and close-up details for the surfaces reconstructed using each of the presented methods as well as for the ground-truth surface.

**Drill dataset.** The drill dataset is the smallest one in the comparison. However, the relatively low sampling density of the surface makes this dataset challenging. In Fig. 9.1, it is shown how the marching cubes methods –RIMLS-MC and Poisson-Rec– fail to reconstruct the original, due to over-smoothing introduced by the regularization mechanism. BPA clearly shows topological errors, whereas ReOP creates an excessive number of faces, which corresponds to its condition of not guaranteeing topological correctness. The proposed method provides an almost-exact replica of the original surface.

**Bunny dataset.** This is a classical, common dataset in the literature that presents a mostly regular sampling with some holes on its base and irregular sampling on the seams between different scans. In Fig. 9.2, the detail view corresponding to the tail
Figure 9.1: First row: general view of the surface of the Drill dataset, from ground-truth – first column – and each of the compared techniques. All of them output the same number of vertices in the reconstructed surface. Second row: detailed view of the surface, with visible edges, corresponding to the top of the drill.
shows how the proposed method and BPA provide a meshing very close to that of the ground-truth. ReOP, once again, generates an excessive amount of triangular faces, whereas RIMLS-MC presents some holes and Poisson-Rec oversmoothes the resulting surface.

**Armadillo dataset.** The *armadillo* dataset introduces a greater level of detail than the previous two datasets. It is also a common dataset in the literature that presents a very regular sampling with a somewhat more complicated topology on the surface corresponding to the teeth of the figurine. Fig. 9.3 shows the reconstructed surfaces obtained with each of the compared methods. The detail views of the proposed method, BPA and RIMLS-MC reveal different polygon meshes, which closely resemble the one corresponding to the ground-truth. ReOP presents an excessive number of triangular faces, whereas Poisson-Rec over-smoothes the reconstructed surface.

**Hand dataset.** The *hand* dataset shows a challenging scenario with lots of joints between different pieces of the complete surface and hollow structures enclosed by two layers of opposite-oriented surfaces. However, the surface lacks fine detail, which makes it a good dataset for marching cubes-based techniques. In Fig. 9.4, general views already show some errors in BPA and ReOP, due to the small distance between the mentioned two-layers of opposite-oriented surfaces. The proposed method and RIMLS-MC provide the most accurate surfaces in topological terms, whereas Poisson-Rec tends to over-connect different pieces due to the excessive volumetric smoothing.

**Dragon dataset.** The *dragon* dataset shows a challenging scenario with a very irregular sampling and a great level of detail on its shape. In Fig. 9.5, the general views reveal that all methods provide apparently hole-free surfaces. However, a closer inspection of the back of the head shows that the proposed method and BPA are the ones that most closely resemble the ground-truth mesh. ReOP generates an excessive amount of faces and RIMLS-MC presents some small-scale holes, whereas Poisson-Rec presents an over-smoothed, yet watertight, output mesh.

**Happy dataset.** The *happy* dataset shows a very challenging scenario with the most irregular sampling and a great level of detail on its shape. In Fig. 9.6, the general views of the reconstructed surface already reveal a loss of detail with RIMLS-MC and Poisson-Rec, due to excessive regularization. BPA presents some cracks in the base, whereas ReOP produces an excessive amount of faces due to the lack of topological guarantees provided by the method. The surface provided by the proposed technique closely resembles that corresponding to the ground-truth.
Figure 9.2: First column: general view of the surface of the Bunny dataset, from ground-truth –first row– and each of the compared techniques. All of them output the same number of vertices in the reconstructed surface. Second column: detailed view of the surface, with visible edges, corresponding to the tail of the bunny.
Surface Interpolation Results

Figure 9.3: First column: general view of the surface of the *Armadillo* dataset, from ground-truth—first row—and each of the compared techniques. All of them output the same number of vertices in the reconstructed surface. Second column: detailed view of the surface, with visible edges, corresponding to a close-up of the face of the figurine.
Figure 9.4: First column: general view of the surface of the Hand dataset, from ground-truth –first row– and each of the compared techniques. All of them output the same number of vertices in the reconstructed surface. Second column: detailed view of the surface, with visible edges, corresponding to the joins between three pieces close to the fifth finger.
Figure 9.5: First column: general view of the surface of the Dragon dataset, from ground-truth –first row– and each of the compared techniques. All of them output the same number of vertices in the reconstructed surface. Second column: detailed view of the surface, with visible edges, corresponding to the back of the head.
Figure 9.6: General and detailed—visible edges—views of the surface of the Happy dataset, for ground-truth and each of the compared techniques. All of them output the same number of vertices in the reconstructed surface. The detailed views show a close-up of the face, where marching cubes-based methods lack fine detail and ReOP clearly introduces unnecessary redundant faces, as well as topological errors.
In general, it can be observed that the proposed method adapts well to all of the datasets and presents a reconstructed mesh which, topologically, is very close to the ground-truth one, apparently delivering better results than the rest of proposed methods. Considering the results from the quantitative comparison above and these qualitative results about the topological correctness of the surface, the proposed technique compares favorably to all of the other tested methods.

9.4 Conclusions

In this chapter, both quantitative –accuracy, computation time and memory footprint– and qualitative results –visual similarity of the reconstructed mesh to that of the ground-truth– have been presented. For obtaining them, the surface interpolation technique proposed in the previous chapter as well as other state-of-the-art approaches, both propagation-based and marching cubes-based have been compared using the same datasets. In general, the main advantages of the proposed technique are the following:

• The proposed method is fast when compared to both propagation-based and marching cubes-based techniques. This is due to the efficient query for surface samples using the \( kd \)-tree structure.

• The proposed method, being part of the propagation-based techniques, offers a better accuracy than marching cubes-based counterparts, since the position of existing surface samples directly translates into the position of the vertices of the resulting mesh.

• When compared to the other propagation-based techniques, the topological correctness of the reconstructed surface obtained with the proposed method makes it a better alternative than the fast, accurate \textit{Restricted and Oriented Propagation} method, the hardest contender in terms of computation time.

• By visual inspection, the mesh resulting from the proposed method resembles the most that of the ground-truth.

About its drawbacks, the most relevant one is the fact that, as a difference with marching cubes-based methods, in which a volumetric, global regularization takes place, when the input data is noisy, so is the output surface. However, this problem might be tackled by a pre-processing stage of the input surface samples, such as the surface smoothing that has been presented in Section 6.5.2, in the chapter devoted to statistical surface sampling.
Part III

Overall Results and Conclusions
Chapter 10

Overall Method: Sampling and Interpolation Results

In the two main parts in which the dissertation is divided, several (progressively improved) approaches for surface sampling and one for surface interpolation have been presented. In this chapter, the best surface sampling approach (the one presented in Chapter 6) and the surface interpolation technique presented in Chapter 8 are concatenated in order to generate 3D reconstructions out of several multi-view video sequences.

The chapter is divided in two parts. The first one presents an evaluation of the proposed approach by comparing it to a baseline system for scene reconstruction composed of classical techniques. The second part presents results obtained from multi-view video sequences publicly available from [4D Repository, 2010], with datasets considering different numbers of input views. These experiments have been used in order to demonstrate the achievement of the main goals of this thesis. As a remainder, the surface reconstruction was expected to be fast –towards real-time–, effective –usable for both analysis and visualization– and efficient –able to compress the multi-view data redundancies–, and the experiments corroborate the accomplishment of these requirements.

10.1 Evaluation

In this first section we present experimental evidence showing that the system resulting from the concatenation of surface sampling and interpolation obtains fast, effective and accurate descriptions of surfaces when compared to a baseline system built from existing techniques.
10.1.1 Data and methodology

Two types of multi-view sequences are used for this first set of experiments. The first one consists of the projected silhouettes of an animated 3D model of a moving person onto a set of 8 cameras. Using such a synthetic sequence is useful for our experiments, since the 3D model used to generate it is available and can be used as 3D ground-truth, which in general is not available in real sequences. The second sequence consists in real data of a moving person captured by 18 cameras in the facilities of Telefonica R&D and the corresponding automatically obtained foreground silhouettes, which present some segmentation errors. The configuration details of these two data-sets are summarized in Table 10.1.

Baseline system. Both the baseline system and the proposed concatenation of surface sampling and interpolation (Table 10.2) use the same silhouette extraction method [Gallego and Pardàs, 2010] for generating their input silhouettes for the real sequences –for the synthetic sequence, silhouettes are noise-free and generated through OpenGL–. Then, in the baseline system, a voxelized shape-from-silhouette implementation that uses projection look-up tables provides a volumetric reconstruction. This implementation technique reduces computing time at the expense of an increased memory usage. At the last stage, a polygon mesh is extracted from the volumetric reconstruction by applying the marching cubes (MC) algorithm. The MC implementation also exploits precomputed tables to improve its throughput. The use of precomputed tables in each stage in order to improve its speed draws a challenging scenario for the comparison.

10.1.2 Experiments

For this evaluation, three experiments have been run on a system equipped with an Intel Xeon 3.0 GHz processor and 32 GB of RAM. These three experiments, which

<table>
<thead>
<tr>
<th>System</th>
<th>Reconstruction</th>
<th>Mesh extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Voxelized SfS</td>
<td>Marching cubes</td>
</tr>
<tr>
<td>Proposed</td>
<td>Statistical Sampling</td>
<td>Proposed meshing</td>
</tr>
</tbody>
</table>

Table 10.2: Features of the baseline system vs. those of the proposed approach
Table 10.3: Quantitative comparison between the baseline system and the proposed sampling-interpolation concatenation

<table>
<thead>
<tr>
<th># Points</th>
<th>Baseline</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (s)</td>
<td>RMS×10^6</td>
</tr>
<tr>
<td>5000</td>
<td>0.5</td>
<td>1006</td>
</tr>
<tr>
<td>10000</td>
<td>1.2</td>
<td>836</td>
</tr>
<tr>
<td>20000</td>
<td>3.0</td>
<td>752</td>
</tr>
<tr>
<td>40000</td>
<td>12.9</td>
<td>673</td>
</tr>
</tbody>
</table>

provide both quantitative and qualitative assessments about the performance of the methodology presented in this thesis, are the following:

- The first one is a performance comparison between both systems, using multi-view sequences with 3D ground-truth data.
- Then, in the second experiment, surface reconstruction results obtained with both systems are qualitatively compared by visual inspection.
- Finally, in the third experiment, a possible Free-Viewpoint Video [Miller et al., 2005] application with the proposed technique is presented, in which a 3D reconstructed subject is placed in a virtual environment with a virtual camera that can be manually placed.

Quantitative performance comparison

The first experiment compares the performance of both systems, baseline and sampling-interpolation concatenation. This comparison consists in the reconstruction of the surface of the animated 3D model and its posterior comparison to the original mesh using the RMS Hausdorff distance [Aspert et al., 2002] between the reconstructed mesh and the real one, averaged along the whole sequence, as a quality measurement. As shown in Table 10.1, no color information is available in the synthetic sequence.

Table 10.3 summarizes the average times employed in the concatenation of 3D reconstruction—statistical surface sampling or volumetric shape-from-silhouette—and meshing—proposed surface interpolation method or marching cubes—and the average RMS Hausdorff distance for both the baseline system and the proposed sampling-interpolation scheme with growing sampling densities. In the table, lower values of RMS mean higher quality. The proposed system offers more accurate results at every resolution level, due to the adopted sampling scheme, better adapted to surfaces. At higher resolution levels, much smaller processing times and better accuracy reflect the suitability of the approach for dense surface reconstruction.
Figure 10.1: Mesh from the baseline system (left) and from the sampling-interpolation concatenation (right) with approx. processing times of about 5 s. Color is added for better discrimination.

Qualitative comparison

The second experiment consists in a qualitative comparison of both systems in a real scenario (using the real sequence). In Fig. 10.1, two meshes corresponding to the surface of a person in a controlled environment, reconstructed by the baseline system and by the sampling-interpolation concatenation, are shown. Both for this experiment and the next one, a conservative estimate of the visual hull with $\tau = 2$, the tolerance to foreground segmentation errors, has been obtained with both methods, due to the notorious presence of foreground misses.

In this case, an equal processing time for both methods (~5s) is the comparison criterion, which results in a larger number of reconstructed points for the proposed technique (30,000 versus 12,000) that, in addition, are more accurate in terms of their location with respect to the ground-truth. Visual inspection reveals that the mesh obtained by the sampling-interpolation concatenation is more accurate and realistic than the one obtained by the baseline system. Furthermore, the meshing algorithm proposed in Part II of this thesis delivers a manifold surface even in challenging regions (Fig. 10.2).

Free-viewpoint video application

In the third experiment, the complete surface reconstruction system—including surface coloring, not used in the previous experiments—has been applied to the real sequence. With this, we can demonstrate a possible free-viewpoint video application, consisting in a replacement of the real background by a virtual one and a free selection of the viewpoint. Please note that the mesh is not textured. Instead, we apply Gouraud shading [Gouraud, 1971] of the triangle faces of the mesh after setting vertex colors equal to those of the corresponding reconstructed surface points.
In Fig. 10.3, two instants of the sequence are viewed from three novel viewpoints.

This possible application reflects the fact that the reconstructed surface contains a suitable amount of information about the foreground elements of the scene. Such information can be employed, for example, for visualizing—from a freely chosen viewpoint—the foreground elements of the scene without the need of using the original images. In other words, the approach proves to be effective for its usage in one of the target applications.

10.2 Validation

In the previous section, it has been shown how the proposed methodology performs faster and with better accuracy than an equivalent system for surface reconstruction built from classic algorithms. In this section, we aim at illustrating how the proposed methodology can obtain high-quality reconstructions of scenes of diverse types. As a result of these experiments, both the effectiveness and the efficiency of the proposed surface reconstruction method for multi-view sequences are showcased.

10.2.1 Data and methodology

For this second set of experiments, fourteen sequences from three datasets—dancer, children and martial—available in [4D Repository, 2010] have been used, which are listed in Table 10.4. As it is shown, the two latter are captured by 16 cameras placed all around the scene, whereas the dancer sequence only contains 8 views. The complete set of multi-view sequences are used for the first experiment, whereas only one sequence of each dataset has been used for the second experiment.
Figure 10.3: Three novel viewpoints (rows) from two different time instants (columns) depicting how a Free-Viewpoint Video application, in which the original background has been replaced by a virtual one, would look like using the methodology presented in this thesis. The scene is captured by 18 cameras and the automatic foreground extraction introduces a notorious presence of segmentation errors. The original multi-view video sequences are courtesy of Telefonica R&D.
### Table 10.4: Datasets from the 4D repository

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Resolution</th>
<th># cams</th>
<th>Sequence</th>
<th># frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dancer</td>
<td>$780 \times 582$</td>
<td>8</td>
<td>dancer</td>
<td>201</td>
</tr>
<tr>
<td>Children</td>
<td>$1624 \times 1224$</td>
<td>16</td>
<td>antoine, antoine assis ballon, antoine edmond ballon, antoine lucie, antoine roue, lucie corde</td>
<td>529, 516, 339, 485, 507, 410</td>
</tr>
<tr>
<td>Martial</td>
<td>$1624 \times 1224$</td>
<td>16</td>
<td>benjamin baton, coup de pied 1, coup de pied 2, parade coup de point 1, parade coup de point 2, saisie col, saisie poigne</td>
<td>233, 210, 240, 241, 256, 310, 321</td>
</tr>
</tbody>
</table>

The video sequences are provided of foreground silhouettes of great quality, with minor errors in few frames. This means that for these experiments it will not be necessary to obtain a conservative estimate of the visual hull, which translates in a more accurate surface reconstruction. Another nice feature of this data is that color is calibrated across views, which is useful for both the application of photo-consistency or surface coloring.

As in the previous section, the sampling-interpolation concatenation presented throughout this thesis has been used for reconstructing the scene at each time instant. The surface sampling stage consists in the statistical sampling (Chapter 6), whereas the surface interpolation stage consists in the application of the method presented in Chapter 8. The number of surface samples in the first stage, unless otherwise stated, has been set to 100000 and the different distances parameterizing the method are set to the same values as those in Table 7.6. For the meshing stage—surface interpolation—, a number of 20 nearest-neighbors is chosen for surface propagation, which results in a closed surface without an excessive computational cost ($\sim 2$ s/frame).

**10.2.2 Experiments**

The fourteen videos have been processed by the sampling-interpolation concatenation, and the resulting surfaces have been re-projected onto the original viewpoints. The experiments in this section show how the proposed methodology is effective
Overall Method: Sampling and Interpolation Results

–suitable for both analysis and visualization–:

- It is able to retrieve with a suitable level of precision the original images by composing the projection of the reconstructed surfaces with the original background or, in other words, it contains most of the information contained in the foreground of the original images.

- The reconstructed surface can also be used for visualization, since it is closed and contains the texture information –as already seen in the first part of this chapter–.

Using the same data, it is also shown that the presented methodology is efficient, since under acceptable level of losses, it is able to compress the multi-view data for video sequences.

Effectiveness

The effectiveness of the methodology presented in this thesis is showcased by presenting two types of qualitative results: the resemblance between the re-projection of the reconstructed surfaces and the original views and the comparison of the reconstructed surface with that obtained with a state-of-the-art method, the Exact Polyhedral Visual Hull (EPVH) [Franco and Boyer, 2003]. For this experiment, the complete set of multi-view sequences of each dataset has been used.

Dancer dataset. In Fig. 10.4, some sampled time instants of the results obtained using the dancer sequence are shown. In this figure, some of the original input images are shown next to novel images generated by composing the re-projection of the reconstructed surface with the original image in order to add the missing background information in 3D –separate background was not available–. The surfaces reconstructed with the proposed method –with and without surface colors– and with EPVH are also shown at the bottom of the figure, seen from a novel virtual viewpoint.

As it can be observed at the top of the figure, the resulting surface, once projected onto the original views, presents some artifacts, due to the conservative method to determine surface sample visibility in the coloring stage and, most importantly, the type of reconstruction –silhouette-consistent–. However, the reconstructed surface still accurately describes the surface of the actual dancer up to a point where it closely resembles the original views after re-projection.

As it is shown at the bottom, the sampling density is high enough to represent with an acceptable level of precision the texture of the face of the dancer and some
Figure 10.4: Dancer (8 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). Some artifacts are visible, e.g. under the arm in the top right view. Bottom: reconstructed surfaces (a) EPVH; (b) proposed and (c) proposed with color. The proposed method is better oriented to represent per-vertex features thanks to the dense surface sampling.
Overall Method: Sampling and Interpolation Results

details in her dress. This contrasts with the reconstruction provided by the EPVH, in which most of the triangular faces are of great dimensions, because it is not oriented to represent per-vertex color information.

Children dataset. At the top of Figures 10.5, 10.6, 10.7, 10.8, 10.9 and 10.10, some sampled time instants of the sequences of the Children dataset are shown. As with the previous dataset, some of the original input images are shown next to novel images generated by composing the projection of the reconstructed surface onto the original image in order to add the background information. At the bottom of these figures, both the surfaces reconstructed with the proposed method and EPVH are also shown.

As a comment about the reconstructed surfaces, the specular reflections at the top of the red ball in Figs. 10.5 or 10.8 are lost due to the color average that takes place in the coloring stage (Section 6.5.3 from the chapter devoted to statistical surface sampling). One limitation of the proposed method is shown in Fig. 10.10. Due to the specific geometry of the string, which does not present a clear orientation for its surface—it constitutes an almost one-dimensional body immersed in 3D space—it is unlikely that its surface can be reconstructed by the statistical surface sampling method.

Martial dataset. As with the two previous datasets, at the top of Figs. 10.11, 10.12, 10.13, 10.14, 10.15, 10.16 and 10.17, some results for sampled time instants of the sequences in the Martial dataset are shown. At the bottom of each figure, the surfaces corresponding to some of these sampled time instants are seen for novel virtual viewpoints.

Again, a problem related to the geometry of the surface arises in Fig. 10.11. Although this time most of the surface of the stick held by the performer is reconstructed, it clearly shows that the sampling strategy does not adapt properly to these structures. In the other sequences, the behavior of the method is such that the re-projected surfaces closely resemble the original images, although they lack specular reflections and present some coloring artifacts due to the high resolution of the input images and the relatively sparse surface sampling.

Overall, the experiments presented in this section are useful as a validation of the effectiveness of the proposed approach. Indeed, at its output, the proposed methodology is able to accurately represent the surfaces of the foreground objects in multi-view scenes, including color information. This representation is useful for visualization, as demonstrated by the projection of the reconstructed surfaces onto novel viewpoints shown at the bottom of each figure.

Furthermore, since the information about these surfaces is presented as a single
Figure 10.5: *Antoine*, from the *Children* dataset (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). There are some visible artifacts (*e.g.* the missing specular reflections on the top of the ball in the middle right image). Bottom: reconstructed surfaces (a) EPVH; (b) proposed and (c) proposed with color.
<table>
<thead>
<tr>
<th>Frame</th>
<th>Original</th>
<th>Overlay</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><img src="image1.png" alt="Frame 0 Original" /></td>
<td><img src="image2.png" alt="Frame 0 Overlay" /></td>
</tr>
<tr>
<td>257</td>
<td><img src="image3.png" alt="Frame 257 Original" /></td>
<td><img src="image4.png" alt="Frame 257 Overlay" /></td>
</tr>
<tr>
<td>515</td>
<td><img src="image5.png" alt="Frame 515 Original" /></td>
<td><img src="image6.png" alt="Frame 515 Overlay" /></td>
</tr>
</tbody>
</table>

**Figure 10.6:** *Antoine assis ballon* (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). The most visible artifact is in the transition between the ball and the left leg in the bottom right image. Bottom: composition of reconstructed surfaces in the three time instants for (a) EPVH; (b) proposed and (c) proposed with color.
### 10.2 Validation

#### Figure 10.7: Antoine Edmond ballon (16 views)

Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). Some artifacts are visible, e.g. the missing color at the bottom of the ball in the bottom right image, or the missing specular reflection on the top of the ball. Bottom: reconstructed surfaces (a) EPVH; (b) proposed and (c) proposed with color.

<table>
<thead>
<tr>
<th>Frame</th>
<th>Original</th>
<th>Overlay</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>169</td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>338</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
</tbody>
</table>

(a) (b) (c)
<table>
<thead>
<tr>
<th>Frame</th>
<th>Original</th>
<th>Overlay</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><img src="image" alt="Frame 0" /></td>
<td><img src="image" alt="Frame 0 Overlay" /></td>
</tr>
<tr>
<td>242</td>
<td><img src="image" alt="Frame 242" /></td>
<td><img src="image" alt="Frame 242 Overlay" /></td>
</tr>
<tr>
<td>484</td>
<td><img src="image" alt="Frame 484" /></td>
<td><img src="image" alt="Frame 484 Overlay" /></td>
</tr>
</tbody>
</table>

**Figure 10.8:** Antoine Lucie (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). The most noticeable artifacts are the missing specular reflections at the top of the ball in the right images and the loss of geometric detail around the chest of Antoine. Bottom: reconstructed surfaces (a) EPVH; (b) proposed and (c) proposed with color.
Figure 10.9: Antoine roue (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). The most visible artifact is the loss of detail on the orange hood in the bottom right image. Bottom: composition of reconstructed surfaces in the three time instants (a) EPVH; (b) proposed and (c) proposed with color.
Figure 10.10: Lucie corde (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). A clear artifact is the loss of the string, which is masked in the top views by the composition with the original images. Bottom: reconstructed surfaces (a) EPVH; (b) proposed and (c) proposed with color. The sampling strategy does not adapt well to the geometry of the string.
10.2 Validation

Figure 10.11: Benjamin baton, from the Martial dataset (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). Some artifacts are visible: e.g. the missing colors on the chin in the bottom right image. Bottom: reconstructed surfaces (a) EPVH; (b) proposed and (c) proposed with color. As in Fig. 10.10, the sampling strategy does not adapt well to the geometry of the stick.
Overall Method: Sampling and Interpolation Results

Figure 10.12: Coup de pied 1 (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). The most visible artifact is the wrong coloring of the chest of the standing man in the bottom right image. Bottom: composition of reconstructed surfaces in two time instants (a) EPVH; (b) proposed and (c) proposed with color. As seen below, 100000 samples are in the sparse side with this dataset.
**Figure 10.13:** *Coup de pied 2* (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). A visible artifact is the lack of detail around the belt of the man at the left side of the bottom right image. Bottom: composition of reconstructed surfaces in two time instants (a) EPVH; (b) proposed and (c) proposed with color. Again, 100000 samples are sparse for representing all the detail.
Figure 10.14: Parade coup de point 1 (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). Some artifacts are visible, e.g. the missing detail on the hair of the man getting up in the bottom right image. Bottom: composition of reconstructed surfaces in two time instants (a) EPVH; (b) proposed and (c) proposed with color.
Figure 10.15: Parade coup de point 2 (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). The most visible artifact is the lack of detail around the belt of the man facing the camera in the top right image. Bottom: reconstructed surfaces (a) EPVH; (b) proposed and (c) proposed with color. The face of the man looking down has not been completely colored due to its orientation.
Figure 10.16: Saisie col (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). A visible artifact is the lack of resolution on the face of the turning man in the bottom right image. Bottom: composition of reconstructed surfaces in two time instants (a) EPVH; (b) proposed and (c) proposed with color. The reconstruction from EPVH cuts the head of the man in one of the time instants.
<table>
<thead>
<tr>
<th>Frame</th>
<th>Original</th>
<th>Overlay</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>160</td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>320</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 10.17: Saisie poigne (16 views). Top: comparison of three original images (left) and the overlay of the reconstructed surface over the original background (right). Some artifacts are visible: e.g. the missing detail on the hair of the man facing the camera in the bottom right image. Bottom: composition of reconstructed surfaces in two time instants (a) EPVH; (b) proposed and (c) proposed with color.
instance, containing both geometry—topology and orientation—and color information, it can also be used for further analysis. The correctness of the surface topology obtained with the meshing approach presented in Part II of this thesis has proven useful in order to correctly interpolate the surface color between samples.

**Efficiency**

The efficiency of the methodology presented in this thesis is also based on the suitable representation of highly redundant data from multi-view settings as a unique instance, containing all the relevant information about the foreground objects. This is possible when the background is known and static, as occurs in the sequences presented for the experiments in this section.

The main idea behind this experiment is to compare the transmission or storage resources that are required to represent the relevant information of multi-view sequences when either an image-based representation of the scene or a 3D model of the foreground are chosen for representing the multi-view data.

The first question that arises when trying to compare the resources required by either approach is how to choose a reasonable surface sampling resolution given a set of multi-view video sequences. A possible answer to such question, which is the one we apply here, is to choose a surface sampling density such that an increase of such density does not produce a significant gain in accuracy.

**Accuracy.** In order to quantify the accuracy, we use the PSNR between the re-projected surface onto the original viewpoints and the foreground pixels of the
original images, which are the only ones used for reconstruction. In the sequences used for our experiments, we have found that the most limiting view in terms of PSNR is that in which the foreground objects occupy most space in the image. This seems reasonable, since more samples are expected to be required in order to accurately represent the contents of a larger number of pixels.

In Fig. 10.18, the average PSNR in the most limiting view of the dancer sequence, with increasing number of surface samples, is shown. Equivalently, in Fig. 10.19 and Fig. 10.20, the average PSNR of the antoine edmond ballon sequence from the children dataset and that of the saisie col sequence from the martial dataset are shown, respectively.

In all cases we observe that the surface reconstruction system reaches a limit of PSNR for a certain sampling density (≈50 dB for the sequences from the dancer and martial datasets and ≈70 dB for the sequence from the children dataset), beyond which little variation in the average PSNR occurs. In some cases it can even decrease. We attribute this to the reconstruction of regions corresponding to shadows for certain sampling densities.

In Fig. 10.21, the re-projection on the most limiting view of the surfaces obtained from a frame in each sequence are shown for increasing number of surface samples. As suggested by the charts, given a sufficient surface sampling, further increasing the sampling density does not produce a significant gain in re-projection accuracy.

For the sequence from the dancer dataset, a number of 20000 surface samples (between the second and third row in Fig. 10.21) already reaches a reasonable level of accuracy for the most limiting image in the multi-view set. For the sequence
from the children dataset, a number of 50000 surface samples (third row in the same figure) is already approaching the limiting accuracy, whereas for the sequence from the martial dataset, 200000 surface samples (fourth row) retrieves the right level of detail.

**Sampling density.** In order to put these figures in a common framework, we normalize both the average PSNR by its maximum value and the number of surface samples by the maximal number of pixels in the most limiting view of each of the processed sequences and show the resulting evolutions in Fig. 10.22.

As shown in the chart, a reasonable choice for the number of surface samples in the sequences used for these experiments would be approximately equal to the number of foreground pixels in the most limiting view. However, in order to be on the safe side, a more conservative, yet still economical, choice would be to take a number of surface samples of approximately twice as many as foreground pixels in the most limiting view –that with the largest silhouette–.

This choice seems also reasonable when we think about an approximately symmetric foreground object –like a cylinder– with a very small scale texture covering its surface. In such a scenario, approximately half of the surface samples would project onto the most limiting view, since half the complete surface would be hidden by the visible half. In that case, in order to retrieve the small scale texture by re-projection of the samples, a sampling density of at least one visible surface sample per pixel would be required, assuming a uniform distribution of the projection of the samples. Since most of the surfaces in natural scenes are expected to have less detail than the assumed in this example, a total number of surface samples of
Figure 10.21: From top to bottom, surfaces reconstructed using 2000, 10000, 50000, 200000 and 1000000 output surface samples. 10% of these samples are obtained as seed samples and the rest through the propagation mechanism in statistical surface sampling.
around twice as many as foreground pixels in the most limiting view is a plausible choice. With this result, we can assume to have an efficient, accurate sampling, with controlled losses, when the rule of thumb presented above is considered.

**Resources.** From the charts in Figs. 10.18, 10.19 and 10.20, a number of surface samples larger than 20000, 50000 and 200000 suffice in order to accurately represent the information contained in the views of the sequences from the dancer, children and martial datasets, respectively. In order to stay in the safe side, 50000 surface samples are used for dancer, 100000 for children and 500000 for martial.

Using these values, which provide the best accuracy in terms of re-projection while using a slightly over-dimensioned amount of resources, a fair comparison can be made about the resources needed to store or transmit the multi-view sequences with a classical image-based manner and with the representation proposed in this thesis, which still introduces losses. This comparison highlights the efficiency of the proposed method, but it does not pretend to introduce the methodology presented in this thesis as a coding mechanism – e.g. [Mueller et al., 2009].

In case of representing the multi-view streams of the sequence from the dancer dataset as sets of lossless PNG images, a total of 808 MB need to be stored (or transmitted), without considering the additional capacity required for the foreground silhouettes. In contrast, a surface-based representation where a set of images containing the static background is stored (transmitted) only once as a set of PNG
10.2 Validation

<table>
<thead>
<tr>
<th>Representation</th>
<th>Compression</th>
<th>dancer</th>
<th>children</th>
<th>martial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image-based</td>
<td>PNG frames</td>
<td>808 MB</td>
<td>12 GB</td>
<td>27.2 GB</td>
</tr>
<tr>
<td></td>
<td>H.264 lossless</td>
<td>352 MB</td>
<td>4.6 GB</td>
<td>4.3 GB</td>
</tr>
<tr>
<td>Surface-based</td>
<td>FG Uncompressed</td>
<td>756 MB</td>
<td>2.4 GB</td>
<td>12.1 GB</td>
</tr>
<tr>
<td>(PSNR)</td>
<td>FG OpenCTM</td>
<td>76 MB</td>
<td>242 MB</td>
<td>1.2 GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50.72 dB</td>
<td>70.85 dB</td>
<td>48.40 dB</td>
</tr>
</tbody>
</table>

Table 10.5: Comparison of the data required to represent multi-view sequences in a classical image-based manner—either as individual frames or exploiting temporal redundancies as lossless video sequences—and using the approach presented in this thesis, keeping the background as a set of PNG images and exploiting spatial redundancies in order to represent the foreground images, and the complete geometry and color of the foreground elements is represented in a compact form for each frame—without compression—, requires 756 MB.

The gain in this case is not dramatic, due to the relatively sparse set of views (only 8) and the low resolution of the input images. However, lossless compressing the geometry using the open source mesh file format OpenCTM [Geelnard, 2010], the latter figure shrinks to just 76 MB. This does not only introduce a clear gain with respect to the former figure, but also when compared to the introduction of lossless video coding—H.264 lossless—, where the initial 808 MB are reduced to just 352 MB by the exploitation of temporal correlation.

In the other two sequences, where the number of views (16) and the image resolution are both larger, further shrinkage of storage or transmission resources are achieved. All these figures are summarized in Table 10.5. The main conclusion that can be extracted from the presented figures is that the introduction of losses with respect to the original images due to the reconstruction process—silhouette-consistent in this case—is counter-balanced by the efficiency of the representation in terms of required resources.

This last experiment showcases the suitability of the proposed approach in terms of its efficiency. It clearly provides a suitable representation of the multi-view data by exploiting the spatial redundancies in the original views, which could still be enhanced by exploiting temporal redundancies using a dense motion estimation as the one presented in Chapter 5. This efficiency justifies its potential for real-time analysis and visualization applications.
10.3 Conclusions

In this chapter, two different types of experiments have been presented, which validate the multi-view reconstruction approach followed in this thesis, based on the division of the surface reconstruction task in two stages: the first one devoted to the sampling of the surfaces of foreground objects and the second one devoted to their interpolation in order to retrieve continuous surfaces. These two types of experiments, which are grouped in two sets of results, are:

- A comparison to a baseline system, in order to characterize the speediness and effectiveness –usability of the reconstructed surfaces– of the proposed approach, using both synthetic and real data.


The first set of experiments has shown how the proposed methodology permits obtaining more accurate reconstructions faster, with smaller computing times when compared to a system composed by classical techniques for volumetric reconstruction –voxelized shape-from-silhouette– and surface extraction –marching cubes–. This reflects the suitability of the design decision of directly reconstructing the surfaces with a set of conveniently designed algorithms in contrast with volumetric techniques, which spend a large part of their computational resources in regions not containing surfaces.

The last part of this first set of experiments also presents, qualitatively, the idea that the proposed representation is effective, for it can be used for a potential free-viewpoint video application. This idea is followed by the first part of the second set of experiments, where the proposed methodology shows its capability for recovering the contents of the original images with a high degree of fidelity by re-projecting the surface reconstructions of the foreground objects and adding background information. It also shows that the quality of the reconstructed surface can rival with that of a state-of-the-art technique (EPVH).

The last experiment in the second set demonstrates the efficiency of the resulting data representation for multi-view video sequences with known and static background. Accepting a reasonable level of losses, due to the reconstruction process –silhouette-consistent–, the resulting surface-based representation greatly reduces the required amount of computational resources for storing or transmitting the multi-view data. This could still be improved by exploiting temporal redundancies in order to represent the foreground –moving– elements in multi-view sequences.
As a result of the presented experiments, we have also derived a rule of thumb for the required surface sampling density in order to represent the information contained in the original images with minimal losses: a reasonable number of surface samples is approximately twice the number of foreground pixels in the view with the largest foreground silhouette.
Chapter 11

Conclusions

The main goal of this thesis, summarized in a single sentence, was to obtain a fast –towards real-time–, efficient –within a compressed/limited support–, effective –exploitable by forthcoming applications– and unique surface-based representation of each frame in multi-view video sequences, which provides all the relevant information contained in the original streams and can be used for both interactive visualization and analysis applications.

In order to obtain such surface-based representation, a strategy consisting in dividing the surface reconstruction task in two stages has been chosen. First, one of the surface sampling strategies proposed in the first main part of the dissertation manages to find a dense cloud of surface points with additional information, such as orientation or color. Next, a fast meshing algorithm is responsible for obtaining the closed, continuous surface out of the samples, which can be used to interpolate the sampled surface features, let them be color, texture, orientation, velocity or any other features of interest.

In this closing chapter, we will first summarize the contributions that resulted from the research on effective, efficient and fast strategies for obtaining surface representations suitable for both analysis and visualization. Then, a number of future lines of work are proposed as either extensions or alternatives to the proposed framework.

11.1 Summary of Contributions

Checking the contents of the two parts of this thesis, different achievements have been made in each of the chapters, effectively contributing to the ultimate goals presented in the introduction in Chapter 1. These contributions basically consist in
the design of methods to obtain the required output representation in every stage—surface sampling and surface interpolation—which are oriented towards real-time applications.

11.1.1 Surface sampling

In Part I, different strategies for sampling the surfaces of objects of interest (foreground objects in scenes with known, static, background) in multi-view settings have been presented, each with its advantages and drawbacks. However, they all fulfill one of the conditions of the proposed thesis, which is that they should be able to work in scenarios with arbitrarily wide baselines. Three different sampling strategies have been presented, comprising one chapter each.

Image-based surface sampling

In the first chapter devoted to surface sampling, an image-based sampling strategy has been presented, which succeeds at obtaining a pixel-wise photo-consistent surface from a set of views in an arbitrarily-wide baseline setup. This approach, which provides an accurate sampling of the visible features and an implicit handling of visibility, has two important drawbacks: on the one hand, it does not provide of closed surfaces; on the other, the strategy for search of surface samples does not provide a natural mechanism to exploit temporal redundancies in multi-view video sequences.

Surface sampling by deformation

In the second presented strategy for surface sampling, the design is driven by the search of a solution to the two main drawbacks of the previous approach. The technique, which consists in a continuous deformation of an existing surface from frame to frame in order to fit the multi-view silhouette constraints, succeeds at providing a closed surface. Not only the reconstructed surfaces are closed, but the proposed technique is also able to track their evolution in sequences, reducing the search area for surfaces in every instant and, therefore, also reducing the computational cost of reconstruction by exploiting temporal correlations.

However, this technique loses the efficiency at imposing photo-consistency of the previous approach and presents a feature that makes it difficult to use in a platform for massive parallelization, such as a GPU, due to the design of its initialization algorithm. Furthermore, it is costly to obtain a high sampling density in its initialization stage when the working space around objects of interest is large.
Statistical surface sampling

In the third chapter devoted to surface sampling, an alternative approach that drives the search for surface samples in a statistical manner is presented. The main advantage of this approach is that, although it initially generates a sparse surface sampling with a low search efficiency in large volumes—scouting—, an efficient search strategy in regions very likely containing surface samples around existing ones provides a fast sampling of surfaces—propagation—. This feature improves this sampling strategy’s ability—compared to that of the one based in the deformation of an existing closed surface—of providing surfaces with a sampling density such that, once re-projected onto the original viewpoints, accurately describe the contents of the original images. In this sense, the algorithm does not only exploit multi-view redundancies, but also spatial correlation on the distribution of surface samples. Furthermore, the algorithm can also benefit from the correlations in the temporal axis. The inefficient initial search for a sparse sampling of the surface is not required from frame to frame. Instead, a search over a suitably smaller range around each sample can be used for scouting. The exploitation of multi-resolution also provides a comparable improvement of the efficiency of the scouting stage.

Even though this technique still presents a limitation when compared to the image-based approach (the application of photo-consistency constraints still requires costly global visibility computations), the algorithm on which it is based is easy to parallelize, which makes it suitable to its application in GPU contexts. In this sense, recent results in a Master’s Thesis, developed by the undergraduate student Marc Maceira and co-directed with my advisor, have shown the advantages of using a surface-oriented, massively-parallel algorithm as the statistical surface sampling in a GPU, resulting in a real-time implementation. Furthermore, this sampling strategy is the fastest among the three presented in this thesis, showing the smallest computation time for a given number of surface samples, especially with a large number of input views, where the proposed approach best shows its potential by the efficient exploitation of spatial correlation.

11.1.2 Surface interpolation

In Part II, a fast strategy for generating a closed, continuous surface out of a dense enough set of surface samples is presented, which results in a triangle mesh that can be used to interpolate the information contained in the surface samples. Compared to other meshing techniques in the state-of-the-art, the proposed algorithm is generally faster and provides better accuracy—in terms of Hausdorff distance to ground-truth surfaces—. It is based on a propagation-based scheme which robustly generates a starting triangular oriented face and iteratively propagates through the
current contour on the set of available surface samples until completely covering the surface.

The technique is efficient, using suitable data structures both for spatial queries –kd-tree– and for topology control –half-edge structure–. The algorithm also shows robustness to uneven distributions of surface samples. The spatial queries provided by the kd-tree, in contrast to the use of volumetric, fixed-size bins with close surface samples, are such that even samples unevenly distributed can be retrieved at the expense of an increase of computational cost by varying the number of nearest-neighbors query results. Thus, the algorithm does not require additional iterations at different scales in order to provide a closed surface.

A limitation of the proposed, propagation-based meshing algorithm exists with respect to another family of existing techniques, based on the definition of a volumetric representation by the application of a certain spatial regularization method and final surface extraction via marching cubes. This limitation is that the location of noisy samples is not corrected by imposing global constraints. However, the reverse of this limitation is that every vertex in the resulting mesh will exactly coincide with an input surface sample. Given a noise-free set of surface samples, this is the most accurate and fastest method to linearly interpolate a continuous surface with the information contained in the samples.

11.1.3 Accomplishments

Regarding the ultimate goals of this thesis as presented in the introduction, the contributions of this thesis can be summarized as follows:

- A unique surface-based representation of the multi-view information data, which is efficient –it compresses the multi-view information–, effective –suitable for analysis and visualization– and accurate –minimizing losses– has been obtained.

- The original images can be approximated –with minimal losses– by re-projection of the unique surface-based representation, which means all the relevant information regarding the original views is kept, with the addition of the 3D structure of the scene.

- Spatial and temporal correlations in multi-view video sequences have been exploited in order to reduce the computation time and make the surface reconstruction strategy suitable for real-time operation for both visualization and analysis applications.

- A triangular mesh that represents a closed surface has been obtained in a usable computation time, as a contrast with many of the techniques in the
state-of-the-art. Such a surface produces a compression of the multi-view data when the static background can be separately treated.

Several publications have resulted, which directly reflect, or are derived from, the achievements related to the methods and the results obtained during the development of this thesis. These are listed in Appendix D.

11.2 Future Work

As stated above, this thesis has succeeded in providing a solution to the problems of defining a methodology to compactly represent multi-view data by approximating the 3D information lost during the imaging process taking place in each camera.

Indeed, the followed approach, consisting in a division of the reconstruction process of the surfaces of foreground objects in multi-view sequences in two stages –surface sampling and surface interpolation– has successfully provided a means to determine the location and orientation of surfaces using a sample-based representation –Part I– and provide topologically correct, closed, continuous surfaces that approximate those of the actual 3D scene –Part II–.

However, some extensions to the methodology proposed in this thesis could be introduced in order to increase the suitability or applicability of the provided 3D data representation to different types of applications, which are out of the scope of this thesis.

11.2.1 Surface sampling

Common to all the surface sampling strategies presented in this thesis, and specially indicated for the image-based surface sampling strategy, it would be interesting to consider a tighter scenario, such as the typical one in Multi-View Stereo (MVS). With small-baseline setups, photo-consistency measurements of higher quality, such as normalized cross-correlation, can be applied when surfaces contain a suitable amount of texture. Combining high-quality photometric measurements with good quality silhouettes by imposing spatial regularization in a variational formulation could provide a higher level of accuracy than the one obtained under the consideration of arbitrarily wide-baseline setups.

Regarding the strategy for surface sampling by deformation, it would be wise to introduce spatial regularization in the tracking part of the algorithm in a more principled manner. This could be done by introducing the meshing of the surface samples in the sampling strategy with the use of MVS in small-baseline scenarios and
without the strong limitation of close-to-real-time performance: once the initial set of high-quality surface samples have been obtained, the surface can be assumed to have the correct topology—which cannot be said from silhouette-consistent surfaces—and subsequent deformations can be applied on the resulting surface by using linear programming [Botsch and Sorkine, 2008], or even non-linear approaches [Botsch et al., 2006] in order to fit the new data constraints imposed by each new frame.

In the statistical strategy for surface sampling, it would be interesting to drive the sampling procedure by the scene contents of interest for a given application. For example, if gesture recognition was to be applied on multi-view data, an enhanced level of detail in the representation of limbs and head could be used to improve the fitting of an existing model—with approximately known pose—to the reconstructed data, using algorithms such as Iterative Closest Point or other variants [Rusinkiewicz and Levoy, 2001]. Such improvement in accuracy could be obtained by favoring the propagation in those areas of interest at the cost of a smaller sampling density over the rest of the surface. It would furthermore conform an interesting showcase of the suitability of the proposed statistical sampling strategy, driven by the exploitation of spatial correlation on the location of surface samples.

11.2.2 Surface interpolation

With respect to the meshing algorithm presented in this thesis, which is used in the surface interpolation stage, it could be sped-up by dividing the meshing of the complete surface into the meshing of several parts and posterior stitching. This approach would better exploit the multi-core architecture of current CPUs, although it would also introduce the problem of correctly stitching the different surface patches reconstructed in each thread without producing topological errors.

11.2.3 Background reconstruction

Last but not least, in the development of this thesis it has not been discussed how to proceed with the background. Indeed, assuming all the information of interest is contained in the foreground of the scene, this part of the image contents—which, in wide-lens cameras can take around 90% of the area—has been deliberately left apart for only using it in order to approximate the original images by re-projection of the reconstructed surfaces.

When the background occludes any of the foreground objects for any of the views, this cannot be ignored and its effect has to be considered in the reconstruction of the dynamic part of the scene. In order to be able to obtain a 3D representation of the background, expensive laser-scan techniques could be used
at a pre-processing stage. However, more flexible methods for reconstructing well-structured large regions without rich texture could also be applied, related to the approach presented in [Furukawa et al., 2009].

11.3 Perspectives

Observing the impact of multi-view systems in the research community, it seems reasonable to imagine a future arrival of systems exploiting the enriched available information of multi-view recording to the mainstream audio-visual industry.

Following the evolution of TV systems –first from black-and-white to color systems, then from color to high-definition systems, now from high-definition to stereoscopic (3D) systems–, a likely step ahead in the future is free-viewpoint video. In order to create realistic applications of this type, it will be necessary to augment the audio-visual transmitted information. Whereas there are on-going works about the efficient transmission of multi-view streams, with this thesis we clearly recall the alternative solution, consisting in representing the dynamic part of the multi-view scene (foreground) by using a unique 3D support, treated separately from the rest of the elements of the scene included in the background.

In the last years, an increasing part of the bi-directional communication has jumped from telephonic calls to video-conferencing, providing an enriched telecommunication experience. Related research on haptic technology attempts to provide a new dimension of realism to this experience, as well as in the field of human-computer interaction (HCI). There is also on-going research in systems for bi-directional video-communication with multi-view systems [3D Presence, 2010, Vision, 2010]. These can also contribute to the realistic perception for the interlocutors by allowing free mobility in a virtual conferencing space.

In order to achieve this, an important challenge will have to be faced by the audiovisual industry: the introduction of multi-view settings as part of mainstream audiovisual setups not only in recording studios, but also in homes. In this aspect, we have already seen how monocular cameras have been added to all types of devices –e.g. laptops, cell phones or video-gaming systems–. Also, one of the most quickly market-penetrating new technology devices in history –Microsoft’s Kinect– has open the door to visual communication increased with depth and advanced HCI to the mainstream.

The next step is to increase the number of sensors in the user space. Although this is a critical step, it has already proven to be accessible in audio applications: starting from the last decade of the 20th century, a large amount of users have switched from stereo reproduction systems to multi-channel ones, despite of the
higher space, cabling and cost requirements of the latter. This shows that, given a sufficient amount of new applications, quality and features of interest for the target user, may be motivating enough to adopt multi-view bi-directional communication.

In this second application field, the restriction on the smallest achievable baseline shows up even more clearly than in the scenarios considered in this thesis. It is in these new scenarios where the segregation of foreground and background parts of the scene is not only convenient, but also necessary in order to successfully process the rather sparse multi-view data that is expected to be available, at least in a near future.

To sum up, we expect to see a growing number of systems exploiting multi-view data, either for communication, analysis for HCI or free-viewpoint television, where some of the concepts in this thesis find an application towards economical, accessible systems for the mainstream audiovisual industry.
Part IV

Appendices
Appendix A

Camera Model

In order to relate the contents of the real world with that of a number of cameras capturing views of it, a parametric model of the camera has to be considered. The parameters of such a model can be obtained by different camera calibration methods, the working principle of which lies beyond the scope of this appendix. In the following, the availability of calibrated parameters is assumed, with a re-projection error smaller than one pixel.

A camera model is generally described by means of two types of parameters, which either model the position and orientation of the camera—extrinsic parameters—or geometric aspects regarding the image creation process of the device—intrinsic parameters. Then, few concepts about back-projection and epipolar geometry are discussed. A more complete introduction to camera geometry can be found in [Hartley and Zisserman, 2000].

A.1 Extrinsic Parameters

The extrinsic parameters describe the orientation and position of a camera with respect to a reference coordinate system. Therefore, this part of the camera model can be properly described by means of a rigid transformation, consisting in a rotation followed by a translation. Such transformation, once applied onto a 3D point described by means of the reference coordinate system, provides the new coordinates of the point referred to a new coordinate system aligned with the camera with its origin at its center of projections—the “camera position”.

Let $\mathbf{X}$ be a 3D point in the reference coordinate system of the real world, $\mathbf{R}$ the axis rotation, $\mathbf{T}$ the translation and $\mathbf{X}'$ the same 3D point referred to the coordinate
system aligned with the camera. Then,

\[ X' = RX + T = \begin{pmatrix} \begin{array}{ccc} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{array} \end{pmatrix} X + \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} \]  \hspace{1cm} (A.1)

Although it could also be equivalently described in reverse order—a translation followed by a rotation—it is more convenient the former option. The reason is that, when using homogeneous coordinates, the previous expression can be compacted to a single matrix operation. Let \( \tilde{X} \) be the point \( X \) in homogeneous coordinates, obtained as

\[ \tilde{X} = \begin{pmatrix} X \\ 1 \end{pmatrix}. \]  \hspace{1cm} (A.2)

Using homogeneous coordinates, Equation A.1 can be written as

\[ X' = \begin{pmatrix} \begin{array}{ccc} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{array} \end{pmatrix} \begin{pmatrix} X \\ 1 \end{pmatrix}. \]  \hspace{1cm} (A.3)

### A.1.1 Camera position

The position of the camera can be described by the position in world coordinates at which its center of projections lies. Let \( X_{CoP} \) be such point and \( X'_{CoP} \) the same point described in the coordinate system aligned with the camera. By construction of the camera-aligned coordinate system,

\[ X'_{CoP} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \]  \hspace{1cm} (A.4)

Then, \( X_{CoP} \) can be obtained as

\[ X_{CoP} = R^{-1} (X'_{CoP} - T) = -R^{-1}T. \]  \hspace{1cm} (A.5)

Using the property that a rotation matrix is orthogonal (\( RR^T = R^T R = I \)), the previous expression can be written as

\[ X_{CoP} = -R^T T. \]  \hspace{1cm} (A.6)

### A.1.2 Camera orientation

The camera-aligned coordinate system has the \( z' \)-axis orthogonal to the image plane, oriented towards the scene. The \( x' \)-axis has the usual horizontal direction on the image plane, oriented from left to right, whereas the \( y' \)-axis corresponds to the
vertical axis on the image plane, oriented from top to bottom. Thus, the camera orientation \( \mathbf{z'} \) equals the third row of the rotation matrix \( \mathbf{R} \):

\[
\mathbf{z'} = \begin{pmatrix} R_{31} \\ R_{32} \\ R_{33} \end{pmatrix}.
\]

(A.7)

### A.2 Intrinsic Parameters

The intrinsic parameters describe the geometry of the imaging process that takes place in the camera. Some models assume the absence of image distortion that appear when using optical lenses, whereas others take this effect into account and model it conveniently.

#### A.2.1 Pinhole camera model

The pinhole camera consists in a small aperture without a lens through which light rays traverse. Therefore, it does not contain optical distortion that appears when using a lens. This model can be conveniently expressed in terms of the calibration matrix \( \mathbf{K} \):

\[
\mathbf{K} = \begin{pmatrix} f \cdot s_x & \lambda & c_x \\ 0 & f \cdot s_y & c_y \\ 0 & 0 & 1 \end{pmatrix}.
\]

(A.8)

In this model, \( f \cdot s_x \) and \( f \cdot s_y \) are the horizontal and vertical focal lengths, respectively, expressed in pixels units. Since most cameras nowadays do not present centering imperfections, the skewness parameter \( \lambda \) can be assumed to be zero. Finally, \((c_x,c_y)\) are the coordinates of the camera’s principal point (ideally at the image center) also in pixel units. With this, the pixel coordinates \( \mathbf{x} = (u,v) \) corresponding to the projection of a 3D point expressed in the camera coordinate system \( \mathbf{X'} \) can be obtained as

\[
\mathbf{z'} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{KX'},
\]

(A.9)

where the natural perspective correction (scaling the pixel coordinates by the point’s depth with respect to the camera \( z' \)) is also applied.

#### A.2.2 Lens distortion

However, real lenses usually have some distortion, mostly radial and slight tangential. Let \( z'\mathbf{X}' = z'(\bar{x}', \bar{y}', 1)^\top \) be the perspective-corrected camera coordinates
before scaling with the calibration matrix $K$. The distorted equivalent, $z'\mathbf{x}'_d = z'(\tilde{x}'_d, \tilde{y}'_d, 1)^\top$, can be obtained as:

$$
\tilde{x}'_d = \tilde{x}'(1 + k_1 r^2 + k_2 r^4 + \ldots ) + 2 p_1 \tilde{x}' \tilde{y}' + p_2 (r^2 + 2 \tilde{x}'^2) \\
\tilde{y}'_d = \tilde{y}'(1 + k_1 r^2 + k_2 r^4 + \ldots ) + p_1 (r^2 + 2 \tilde{y}'^2) + 2 p_2 \tilde{x}' \tilde{y}'
$$

where $r^2 = \tilde{x}'^2 + \tilde{y}'^2$, $k_1$ and $k_2$ are the normalized radial distortion coefficients, and $p_1$ and $p_2$ are the normalized tangential distortion coefficients. The distorted pixel coordinates $\mathbf{x}_d = (u_d, v_d)$ can be obtained as

$$
\begin{bmatrix}
 u_d \\
 v_d \\
 1
\end{bmatrix}
= K
\begin{bmatrix}
 z' \tilde{x}'_d \\
 z' \tilde{y}'_d \\
 z'
\end{bmatrix}
. 
$$

Distortion correction

In order to correct the distortion present in the images captured by a real-world camera—especially those with a wide-angle lens—the image must be warped by the inverse function applied in order to distort the projected coordinates. This can be easily done for every undistorted pixel $\mathbf{x} = (u, v)$ by computing its distorted coordinate $\mathbf{x}_d = (u_d, v_d)$ using Eqs. A.10 and A.11 and taking the bi-linear interpolation of the values in the four closest pixels to $\mathbf{x}_d$ as the value for the undistorted pixel $\mathbf{x}$.

With this, undistorted images are slightly blurred but, in exchange, camera calibration can be used without taking distortion into consideration. The complete set of matrices in order to obtain the image coordinates $\mathbf{x} = (u, v)$, corresponding to the projection a 3D point in homogeneous world coordinates $\mathbf{X}$ onto a camera image with calibration matrix $K$ and pose $(R|T)$ is:

$$
\begin{bmatrix}
 u \\
 v \\
 1
\end{bmatrix}
= K(R|T)\mathbf{X} = P\mathbf{X},
$$

where $P$ is called the projection matrix.

A.3 Back-Projection

Two different types of back-projection of an image point are considered. If the depth $z'$ of the image point is known (range data, stereo, . . .), the exact position of the 3D point originating the image point can be obtained in a straightforward manner, whereas in a more general case where the depth is unknown, the back-projection will result in a ray passing through the camera center and the pixel on the image plane.
A.3.1 Known depth

Given an image point \( x = (u, v) \) with known depth \( z' \), the equivalent point before perspective correction –projective coordinates– is \( z' \tilde{x} = z'(u, v, 1)^\top \). The corresponding 3D point expressed in world coordinates \( X \) with known camera parameters \( K \) and \( (R|T) \) can be simply computed as

\[
X = R^\top (K^{-1}z' \tilde{x} - T) = z'(KR)^{-1} \tilde{x} + X_{CoP},
\]

where \( X_{CoP} = -R^\top T \) is the camera center.

A.3.2 Unknown depth

Given a point \( x \) without known depth in an image and known camera parameters \( K \) and \( (R|T) \), an infinite set of 3D points map to this image point. This set is a ray in 3D space, \( \Gamma_x \), passing through the camera center \( X_{CoP} = -R^\top T \). The other known point of the ray is the image point.

Since in this case the depth is unknown, we define a parametric equivalent point before perspective correction as \( \lambda \tilde{x} = \lambda (u, v, 1)^\top \). Proceeding in the same manner as in the previous case with known depth, we obtain

\[
X(\lambda) = R^\top (K^{-1}\lambda \tilde{x} - T) = \lambda (KR)^{-1} \tilde{x} + X_{CoP},
\]

or, expressed as a set,

\[
\Gamma_x = \{ X(\lambda) = X_{CoP} + \lambda (KR)^{-1} \tilde{x} | \lambda \in \mathbb{R}^+ \}.
\]

In this case, the unknown depth \( z' \) is used as a parameter, \( \lambda \), that produces a back-projected ray with its variation.

A.4 Epipolar Geometry

The epipolar geometry between two views is based on three entities:

- The **epipole** is the point of intersection of the line joining the two camera centers (the baseline) with the image plane. Equivalently, the epipole is the image in one view of the camera center of the other view.
- An **epipolar plane** is a plane containing the baseline.
- An **epipolar line** is the intersection of an epipolar plane with the image plane. As all epipolar planes contain the baseline, all epipolar lines must meet at the epipole.
Let \( \tilde{x}_1 \) and \( \tilde{x}_2 \) be two image points in projective coordinates in cameras 1 and 2, respectively, and let \( l_1 \) and \( l_2 \) be the epipolar lines of \( \tilde{x}_2 \) in camera 1 and of \( \tilde{x}_1 \) in camera 2, respectively. There exists a rank-2 matrix \( F_{12} \) – which is called the fundamental matrix – satisfying

\[
l_2 = F_{12} \tilde{x}_1. \tag{A.16}
\]

By definition, \( \tilde{x} \in l_2 \iff \tilde{x}^T l_2 = 0 \), which results in

\[
\tilde{x}^T F_{12} \tilde{x}_1 = 0, \quad \forall \tilde{x} \in l_2. \tag{A.17}
\]

As a consequence of this, and the symmetry of the problem, it is also true that

\[
l_1 = F_{21} \tilde{x}_2 = F_{12}^{\top} \tilde{x}_2. \tag{A.18}
\]

Related to the back-projection of an image point, line \( l_1 \) corresponds to the image in camera 1 of the back-projected ray of image point \( x_2 \) in camera 2.

### A.4.1 Derivation of the fundamental matrix

Let \( P_1 \) and \( P_2 \) be the projection matrices of cameras 1 and 2, respectively. We denote by \( \tilde{X}_{CoP,1} \) and \( \tilde{X}_{CoP,2} \) the centers of projection of cameras 1 and 2, respectively, in homogeneous world coordinates. Finally, let \( \tilde{e}_1 = P_1 \tilde{X}_{CoP,2} \) and \( \tilde{e}_2 = P_2 \tilde{X}_{CoP,1} \) be the epipoles in projective coordinates in cameras 1 and 2, respectively.

Under these assumptions, we can affirm that the following points belong to the back-projected ray of \( x_1 \):

- The projection center of camera 1: \( \tilde{X}_{CoP,1} \).
- \( P_1^+ \tilde{x}_1 \), where \( P_1^+ = P_1^T (P_1 P_1^T)^{-1} \) is a pseudo-inverse matrix of \( P_1 \), i.e. \( P_1 P_1^+ = I_3 \).

As a consequence, \( P_2 (P_1^+ \tilde{x}_1) \) and \( \tilde{e}_2 = P_2 \tilde{X}_{CoP,1} \) will be two points lying in the epipolar line \( l_2 \). Thus, the epipolar line \( l_2 \) can be computed as

\[
l_2 = \tilde{e}_2 \times P_2 (P_1^T \tilde{x}_1) = F_{12} \tilde{x}_1. \tag{A.19}
\]

So the final expression of the fundamental matrix is

\[
F_{12} = [\tilde{e}_2]_\times P_2 P_1^+, \tag{A.20}
\]

where the cross product has been expressed as a matrix product with a skew-symmetric matrix \( [\tilde{e}_2]_\times \) constructed as

\[
[\tilde{e}_2]_\times = \begin{pmatrix}
0 & -\tilde{e}_{2,z} & \tilde{e}_{2,y} \\
\tilde{e}_{2,z} & 0 & -\tilde{e}_{2,x} \\
-\tilde{e}_{2,y} & \tilde{e}_{2,x} & 0
\end{pmatrix}. \tag{A.21}
\]
Appendix B

kd-Tree

A kd-tree or k-dimensional tree is a space-partitioning data structure for organizing points in a k-dimensional space. It is useful for several applications, such as searches involving a multi-dimensional search key, e.g. range searches or nearest-neighbor searches [Friedman et al., 1977].

The kd-tree is a special case of binary partition tree in which every node is a k-dimensional point. Every non-leaf node can be thought of as implicitly generating a splitting hyperplane that divides the space into two parts, known as subspaces. Points to the left of this hyperplane represent the left sub-tree of that node and points right of the hyperplane are represented by the right sub-tree.

The hyperplane direction is chosen in the following way: every node in the tree is associated with one of the k-dimensions, with the hyperplane perpendicular to that dimension’s axis. So, for example, if for a particular split the x axis is chosen, all points in the subtree with a smaller x value than the node will appear in the left subtree and all points with larger x value will be in the right sub-tree. In such a case, the hyperplane would be set by the x-value of the point, and its normal would be the unit x-axis.

B.1 Construction

Since there are many possible ways to choose axis-aligned splitting planes, there are many different ways to construct kd-trees. The canonical method of kd-tree construction has the following constraints:

- As one moves down the tree, one cycles through the axes used to select the splitting planes. (For example, the root would have an x-aligned plane, the
root’s children would both have \( y \)-aligned planes, the root’s grandchildren would all have \( z \)-aligned planes, the next level would have an \( x \)-aligned plane, and so on.)

- Points are inserted by selecting the median of the points being put into the subtree, with respect to their coordinates in the axis being used to create the splitting plane. (Note the assumption that we feed the entire set of points into the algorithm up-front.)

This method leads to a balanced \( kd \)-tree, in which each leaf node is about the same distance from the root. However, balanced trees are not necessarily optimal for all applications.

Given a list of \( n \) points, the following algorithm will construct a \( kd \)-tree containing those points. The inputs are a list of ordered points and the position of the first and last elements to process in the list. Then, the recursive algorithm goes as follows:
B.2 Queries

1. If the first and last element are the same, return immediately.

2. Create a node, containing the splitting axis and the location of the median in the specified range.

3. Sort the list in the specified range by increasing value in the splitting axis.

4. Recursively create the left sub-tree by calling the function with last := median and keeping the original first.

5. Recursively create the right sub-tree by calling the function with first := median + 1 and keeping the original last.

6. Return the newly created node.

In Fig. B.1, a kd-tree is created by the criterion of maximum-variance splitting axis. This allows obtaining good search performance in sets including outliers.

B.1.1 Dynamic addition of elements

A new point can also be added to a kd-tree after its construction. First, the tree must be traversed, starting from the root and moving to either the left or the right child depending on whether the point to be inserted is on the left or right side of the splitting plane. Once the node under which the child should be located is reached, the new point is added as either the left or right child of this leaf node, again depending on which side of the node’s splitting plane contains the new node.

However, adding points in this manner can cause the tree to become unbalanced, leading to decreased tree performance. The rate of tree performance degradation is dependent upon the spatial distribution of tree points being added, and the number of points added in relation to the tree size. If a tree becomes too unbalanced, it may need to be re-balanced to restore the performance of queries, such as nearest-neighbor searching –introduced below–. A reasonable choice for re-balancing the tree is to construct it from ground, using the current set of points, every time the ratio between the number of dynamic insertions since the last re-balancing and the total number of elements of the tree surpasses a certain threshold.

B.2 Queries

Once the tree is constructed, two types of queries can be processed efficiently, in logarithmic time with respect to the number of elements contained in the tree. The first one, \((k)\)-nearest-neighbors search, is able to find the \(k\) closest points to a given one, whereas the second one, range search, is able to find all the points at a distance smaller than a maximum value from a given point.
B.2.1 Nearest-neighbors search

The nearest-neighbors (NN) algorithm aims to find the points in the tree which are nearest to a given input point. This search can be done efficiently, in logarithmic time, by using the tree properties to quickly eliminate large portions of the search space. Searching for the nearest neighbor in a $kd$-tree proceeds as follows:

- Starting with the root node, the algorithm moves down the tree recursively, in the same way that it would if the search point were being inserted (i.e. it goes right or left depending on whether the point is greater or less than the current node in the split dimension).

- Once the algorithm reaches a leaf node, it saves that node point as the current best.

The algorithm unwinds the recursion of the tree, performing the following steps at each node:

- If the current node is closer than the current best, then it becomes the current best.

- The algorithm checks whether there could be any points on the other side of the splitting plane that are closer to the search point than the current best. In concept, this is done by intersecting the splitting hyperplane with a hypersphere around the search point that has a radius equal to the current nearest distance. Since the hyperplanes are all axis-aligned this is implemented as a simple comparison to see whether the difference between the splitting coordinate of the search point and current node is less than the distance (overall coordinates) from the search point to the current best.

- If the hypersphere crosses the plane, there could be nearer points on the other side of the plane, so the algorithm must move down the other branch of the tree from the current node looking for closer points, following the same recursive process as for the entire search.

- If the hypersphere does not intersect the splitting plane, then the algorithm continues walking up the tree, and the entire branch on the other side of that node is eliminated from the search.

When the algorithm finishes this process for the root node, then the search is complete.

The algorithm can be extended in several ways by simple modifications. It can provide the $k$-nearest neighbors to a point by maintaining the list of the $k$ current
bests instead of just one. Branches are only eliminated when they cannot have points closer than any of the $k$ current bests.

### B.2.2 Range search

As seen in [Preparata and Shamos, 1985], the complexity of finding all points within a given distance of a target point \textit{–i.e.,} performing a range search\textit{–} is still logarithmic in the size of the tree \textit{–i.e.,} the number of points it contains, $n$ \textit{–} for a fixed range distance.

Let $t$ be the target point and $r$ the range distance for the search. Then, this operation can be implemented as a modification of the previous NN algorithm:

1. The distances from the visited points to $t$ are not compared to that of the closest point found, but to the fixed initial value $r$.

2. As in $k$-nearest-neighbors, all the discovered points within this distance are returned in a list, not just the closest.

### B.3 Complexity

Building a static $kd$-tree from $n$ points takes $O(n \log^2 n)$ time if an $O(n \log n)$ sort is used to compute the median at each level. The insertion of a new point into a balanced $kd$-tree takes $O(\log n)$ time.

#### B.3.1 High-dimensional data

Finding the nearest point is an $O(\log n)$ operation in the case of randomly distributed points. However, analysis of binary search trees has found that the worst case search time for a $k$-dimensional $kd$-tree containing $n$ nodes is given by the following equation:

$$ t_{\text{worst}} = O(k \cdot n^{1-1/k}). $$

These poor running times only apply when $n$ is in the order of the number of dimensions $k$. In very high dimensional spaces, the curse of dimensionality causes the algorithm to need to visit many more branches than in lower dimensional spaces. In particular, when the number of points is only slightly higher than the number of dimensions, the algorithm is only slightly better than a linear search of all of the points. As a general rule, if the dimensionality is $k$, then the number of points in the data $n$ should be such that $n \gg 2k$. 
Appendix C

Hausdorff Distance

The Hausdorff distance, named after Felix Hausdorff, measures how far two subsets of a metric space are from each other. It turns the set of non-empty compact subsets of a metric space into a metric space in its own right.

Informally, two sets are close in the Hausdorff distance if every point of either set is close to some point of the other set. Although more details about the Hausdorff metric can be found in [Henrikson, 1999], here we will introduce just some basic notions.

C.1 Definition

The Hausdorff distance \( d_H(X, Y) \) between two non-empty subsets \( X \) and \( Y \) of a metric space \( (M, d) \) is defined as

\[
d_H(X, Y) = \max \{ \sup_{x \in X} \inf_{y \in Y} d(x,y), \sup_{y \in Y} \inf_{x \in X} d(x,y) \}. \tag{C.1}
\]

Equivalently,

\[
d_H(X, Y) = \inf \{ \epsilon > 0; X \subseteq Y_\epsilon \text{ and } Y \subseteq X_\epsilon \}, \tag{C.2}
\]

where

\[
X_\epsilon := \bigcup_{x \in X} \{ z \in M; d(z,x) \leq \epsilon \}, \tag{C.3}
\]

that is, the set of all points within \( \epsilon \) of the set \( X \).

Remark. It is not true in general that, if \( d_H(X, Y) = \epsilon \), then

\[
X \subseteq Y_\epsilon \text{ and } Y \subseteq X_\epsilon.
\]
For instance, consider the metric space of the real numbers $\mathbb{R}$ with the usual metric $d$ induced by the absolute value,

$$d(x, y) := |y - x|, \quad x, y \in \mathbb{R}.$$  

Take  

$$X := \{1/n; n \in \mathbb{N}\} \text{ and } Y := \{-1/n; n \in \mathbb{N}\}.$$  

Then $d_H(X, Y) = 1$. However $X \not\subseteq Y_1$ because $Y_1 = [-2, 1)$, but $1 \in X$.

### C.1.1 Properties

Some properties can be derived from the definition of the Hausdorff distance:

- In general, $d_H(X, Y)$ may be infinite. If both $X$ and $Y$ are bounded, then $d_H(X, Y)$ is guaranteed to be finite.
- We have $d_H(X, Y) = 0$ if and only if $X$ and $Y$ have the same closure.
- On the set of all non-empty subsets of $M$, $d_H$ yields an extended pseudometric.
- On the set $F(M)$ of all non-empty compact subsets of $M$, $d_H$ is a metric. If $M$ is complete, then so is $F(M)$. If $M$ is compact, then so is $F(M)$. The topology of $F(M)$ depends only on the topology of $M$, not on the metric $d$.

### C.2 Motivation

The definition of the Hausdorff distance can be derived by a series of natural extensions of the distance function $d(x, y)$ in the underlying metric space $M$, as follows:

- Define a distance function between any point $x$ of $M$ and any non-empty set $Y$ of $M$ by:

$$d(x, Y) = \inf\{d(x, y) | y \in Y\}.$$  

For example, $d(1, [3, 6]) = 2$ and $d(7, [3, 6]) = 1$.

- Define a distance function between any two non-empty sets $X$ and $Y$ of $M$ by:

$$d(X, Y) = \sup\{d(x, Y) | x \in X\}.$$  

For example, $d([1, 7], [3, 6]) = d(1, [3, 6]) = 2$.

- If $X$ and $Y$ are compact, then $d(X, Y)$ will be finite; $d(X, X) = 0$; and $d$ inherits the triangle inequality property from the distance function in $M$. As it stands, $d(X, Y)$ is not a metric because $d(X, Y)$ is not always symmetric.
and \( d(X, Y) = 0 \) does not imply that \( X = Y \) (It does imply that \( X \subseteq Y \)). For example, \( d([1, 3, 6, 7], [3, 6]) = 2 \), but \( d([3, 6], [1, 3, 6, 7]) = 0 \). However, we can create a metric by defining the Hausdorff distance to be:

\[
d_H(X, Y) = \max\{d(X, Y), d(Y, X)\}.
\]

### C.3 Applications

In feature matching, the Hausdorff distance can be used to find a given template in an arbitrary target image [Alhichri and Kamel, 2002]. The template and target image can be pre-processed via a certain feature extractor giving a set of detected feature points. Next, each detected point in the template image is treated as a point in a set. Similarly, an area of the target image is also treated as a set of points. The algorithm then tries to minimize the Hausdorff distance between the template and some area of the target image. The area in the target image with the minimal Hausdorff distance to the template, can be considered the best candidate for locating the template in the target.

#### C.3.1 Polygon mesh dissimilarity

In computer graphics, the Hausdorff distance is used to measure the difference between two different representations of the same 3D object. Although by definition the Hausdorff distance is the one-sided maximum of the minimum distance, in practice the mean or RMS measures are quite useful, since these are much less sensitive to outliers.

In [Cignoni et al., 1998], a tool is presented, able to compare the difference between a pair of surfaces (e.g. a ground-truth mesh and an estimate obtained from a set of oriented points) by adopting a surface sampling approach. It uses the Hausdorff distance in order to return, among other, maximum and mean error. Recently, this tool has been integrated in the open source mesh editor *Meshlab* [Cignoni et al., 2008].
Appendix D

Related Publications

This appendix contains a list of publications related to the thesis that either have been accepted or are pending acceptance at the moment of finishing the dissertation. These publications are classified by their correspondence to the different part in which this thesis is divided.

D.1 Surface Sampling

Three publications –one corresponding to each proposed sampling strategy– have resulted from the research on efficient methods for extraction of dense sets of samples representing surfaces of foreground objects in scenes with known and static background.

Photo-Consistent Surfaces from a Sparse Set of Viewpoints. In this ICIP 2010 conference article [Salvador and Casas, 2010b], the image-based approach for surface sampling introduced in Chapter 4 is presented as a method capable of obtaining photo-consistent surfaces for foreground objects in arbitrarily wide-baseline scenarios. A comparison with a volumetric approach, also included in the first section of Chapter 7, demonstrates the adequacy of a surface-based approach for 3D reconstruction of the visible features in multi-view settings.

Joint Estimation of Shape and Motion from Silhouettes. In this second ICIP 2010 conference article [Salvador and Casas, 2010a], the strategy for surface sampling based on the deformation of an existing closed surface introduced in Chapter 5 is presented as an efficient technique for obtaining silhouette-consistent surfaces in multi-view video sequences. Besides this result, the possibility of ob-
taining a dense motion field estimate on the surface samples as a simple position
difference between consecutive time instants was also introduced as a possible valu-
able feature for further analysis applications. The experiments presented in this
article are also contained in the second section of Chapter 7.

Statistical Surface Sampling for Multi-View Reconstruction. In the arti-
cle under preparation for Transactions on Image Processing [Salvador and Casas,
2011], the statistical strategy for surface sampling introduced in Chapter 6 is pre-
sented as a valuable tool for efficiently extracting a discrete representation of the
silhouettes of foreground objects that can be used for interactive Free-Viewpoint
Video by using it as input in the meshing algorithm presented in Chapter 8. Some
of the experiments, introducing an efficiency improvement in the scouting stage of
the algorithm –dynamic or multi-resolution–, are reflected in the third section of
Chapter 7. The article will be submitted for review in the next weeks after the
completion of the dissertation.

D.2 Surface Interpolation and Applications

One publication has been made, which introduces the strategy for surface interpo-
lation from a set of oriented surface points.

From Silhouettes to 3D Points to Mesh: Towards Free Viewpoint Video.
In this 3DVP 2010 workshop article [Salvador et al., 2010], the meshing algorithm
presented in Chapter 8 is presented as part of a complete system for efficiently
obtaining surface descriptions suitable for Free-Viewpoint Video applications. Being
the statistical sampling stage still under development, the sampling strategy chosen
for this approach corresponds to the one presented in the first part of Chapter 5 with
the addition of some post-processing stages also used for the statistical sampling
approach. These algorithms are also presented in the last sections of Chapter 6.
The presented results compare the presented surface reconstruction strategy with
a baseline system based on volumetric techniques. Although the differences are not
as great as the ones obtained by using the statistical sampling strategy presented
in the first section of Chapter 10, the experimental results show the adequacy of
the surface-oriented methodology.

Finally, an application for the improvement of foreground silhouettes in multi-
view settings has also been proposed, partially based on some techniques presented
in this thesis.
Joint Multi-view Foreground Segmentation and 3D Reconstruction with Tolerance Loop. In the ICIP 2011 submission [Gallego et al., 2011], an application of the conservative estimate of the visual hull with tolerance to segmentation errors is introduced. The main idea is to iteratively refine noisy foreground silhouettes by introducing *a priori* information from the projections of gradually less conservative visual hulls. In this case, a volumetric representation is chosen in order to introduce spatial regularization using a state-of-the-art energy minimization technique (graph-cuts). Then, the surface extracted using *marching cubes* is fitted to the noisy silhouettes—with tolerance to errors—by using a technique similar to the one presented in the part corresponding to the dynamic sampling of Chapter 5. This application of the method allows to improve the precision of the projection of the reconstructed low-resolution surface. The results show a clear improvement on the quality of the final estimate of the visual hull. The decision upon its acceptance will be announced to the authors in April 2011.

At the moment of finishing this thesis, a new article intended for its publication in *Transaction on Image Processing* is in preparation. This article will present the complete reconstruction approach (surface sampling and interpolation) with new experiments, comparing the proposed technique to state of the art methods like the concatenation of *Patch-based Multi-view Stereo* [Furukawa and Ponce, 2009a, PMVS, 2010] and *Poisson reconstruction* [Kazhdan et al., 2006].


