Exploring Past and Present: VR Reconstruction of the Berlin *Gendarmenmarkt*

Category: Research

ABSTRACT

Virtual reality games demand the use of highly realistic 3D models to provide an immersive environment to the users. Therefor, modeling of real world structures is a common process within the industry. In recent years the photogrammetric tools have become available to aid in this process. However, obtaining high quality imagery is still an issue w.r.t. budget and time constraints. We present a simple, effective method for the reuse of existing video and photo material towards the generation of 3D models suitable for a virtual reality gaming environment. An abundance of existing content may be available in on-line repositories (YouTube, Flickr, Google images,..) and can be exploited. A complete video-to-VR pipeline is presented and a use-case of the Berlin Gendarmenmarkt square is tested, wherein the use of both commercial and open source software is evaluated. Our proposed method improves existing workflows used by VR game designers in both structural accuracy and creation time of real world objects.

1 INTRODUCTION

Digital reconstruction techniques typically operate on newly captured information. However, this only allows the reconstruction of the asset in its current state. While this is the preferred deliverable for most remote sensing applications, it is of great interest to also reconstruct the state of the asset in previous time periods. By doing so, heritage experts are able to visualize and analyze the asset in time, thus effectively adding a fourth dimension to the process. Other stakeholders include game designers that look to reconstruct historically accurate scenery for their games. The paradigm of creating these 4D digital reconstructions heavily relies on the presence of proper data. This is troublesome since most historic data sources were never meant to be used in reconstruction work-flows and thus yield improper results. Also, these data sources are typically heterogeneous and unstructured. It is within the scope of this project to assess already acquired data from sites for their compatibility with current reconstruction work-flows.

We present a use case for virtual reality (VR) game designers to efficiently re-purpose old and new content to aid them during the development process. This use case, called "Living the past", focuses on the reuse of video footage taken from a historic square, the Berlin *Gendarmenmarkt*, in order to create a Virtual Reality environment.

Scenes depicting the square buildings, statues and lamppost were automatically reconstructed as 3D models from a number of video sequences taken from the square. These 3D models were lightly edited to improve their quality and then imported into a Virtual Reality designing tool to be used as basis for the scene production. The Virtual Reality scenes were also enriched with surroundings assets like buses in the present and horse-drawn carriages in the past, to give the impression of different time periods.

The remainder of this work is structured as follows. In section 2, the photogrammetric reconstruction pipeline is proposed. A real test case is shown in section 3. The VR implementation is presented in 4. Finally, the conclusions are presented in Section 5 along with a discussion about future work.

Figure 1: An overview of the *Gendarmenmarkt* square depicting the *Konzerthaus* in the middle. The symmetry of the square is noticeable: the churches to the left and right are nearly identical from several viewpoints. This may pose severe issues for photogrammetric reconstruction [7].

2 PHOTOGRAMMETRIC RECONSTRUCTION

As previously discussed, historic data sets suffer from data heterogeneity. However, low cost photography has been around for a better part of a century and thus each historic structure of significance has been captured with images countless times over the past decades. Our hypothesis is that at least a portion of this information can be used as input for a photogrammetric reconstruction. In the following paragraphs, the work-flow for deriving useful inputs from the available content and fully automatically generating 3D digital models is discussed.

2.1 Available Content

In this work, the Berlin *Gendarmenmarkt* is used as a test case for the photogrammetric reconstruction. This historic market, originating from the 1600's, consists of a cobblestone square surrounded by a number of row houses. A panoramic view is shown in Figure 1. The prominent historical structures include the *Konzerthaus* and the *Schiller* monument statue. Several image sources are considered for the reconstruction. First there are the inputs from separate images. For these we make use of the Europeana Collections database [9], which provides access to over 50 million digitized items - books, artwork, photos, videos and more - all indexed and annotated, making it searchable and filterable. Secondly YouTube was searched for videos of the square released under its Creative Commons license. Other popular on-line repositories such as Flickr, Instagram, ... are also available but were not used in this use-case since this imagery is typically not well inventoried and described. Furthermore, the emphasis of these images lies often more on portraits than on the actual asset, so their value is limited.

In the end, several video and image datasets were available for 3D reconstruction of the *Gendarmenmarkt* from the following sources: Europeana (6 historical images), Deutsche Welle (old video footage), YouTube movies, pictures found on-line and some self-shot videos.

2.2 Previous reconstruction efforts

Previous works have addressed the reconstruction of the *Gendarmenmarkt* [6, 26]. Due to repetitive scene structures, reconstruction was a failure in [26] where a global reconstruction pipeline was employed. This was mainly due to the high resemblance of the German Church (left) and French Church (right), as seen in figure 1.

Figure 2: Various content provided by Europeana Collections (a,b,c) depicting the *Gendarmenmarkt* as early as 1899. Additional content includes publicly available images (d,e), videos (f) and self shot material (g,h).

2.3 Image processing pipeline

In this use case two commercial software packages for 3D reconstruction (Agisoft's Photoscan [1], Capturing Reality's RealityCapture [5]) and our own implementation, based on open source package Colmap [19, 21] have been tested. All three packages are capable of a full photogrammetry pipeline from input images towards 3D model output. Each of these packages employs a slightly different reconstruction strategy. Details of the algorithms of commercial packages are not disclosed by their developers, however some assumptions may be made. The main reconstruction pipeline of all packages is similar.

2.3.1 Frame extraction

Photogrammetric pipelines typically employ still images or frames of a particular scene for 3D reconstruction. Since movies or videos consist of a multitude of frames (typically recorded at 25 or 30Hz) and can contain shots of multiple scenes, they are not directly useful for photogrammetry. Therefor in a preliminary step the various shot are first delineated and can then be further processed to extract proper frames for reconstruction. Since simply decomposing the shots into all of their frames would result in an overload of redundant data, a subset must be made.

FFmpeg [10], an open source library, was used to access the video streams. A threshold-based scene detector was used to segment the shots from each video. To detect the various shots a *scene score* is calculated between consecutive frames. The score is determined by the sum of absolute difference (SAD) between all pixels of consecutive frames. The resulting value varies between 0 and 1 and may be used as a measure for similarity between 2 image blocks. A video cut, and thus a new shot, is assumed once the scene score exceeds a set threshold T_{sad}. All shots were successfully extracted in this use case using $T_{sad} = 0.3$. The different video shots were sampled uniformly to a set of frames by extracting N_f frames per second. Depending on the target object for reconstruction N_f varied between 1 for a reconstruction of the entire square and 5 for the reconstruction of smaller objects or fast camera movement (e.g. statue). The value of N_f was manually determined taking into account the number of output images and the need for sufficient overlap between consecutive frames.

2.3.2 Feature extraction and description

Sparse feature points in the images are extracted and their appearance is described using a numerical descriptor. Widely used and well-performing extraction and descriptor algorithms are SIFT [15] and SURF [4], together with their variants (SIFT-GPU, SURF-GPU, ASIFT, DAISY,..). Feature extraction has a very high influence on the performance and success of the entire pipeline. For this reason all packages employ several settings to limit the number of feature points per image. Due its open source transparency, Colmap allows very in-depth adjustable thresholds and parameters for the feature extraction algorithm. Neither commercial package discloses internal algorithms. Settings here are limited to internal image resizing and the maximum number of feature points per image.

2.3.3 Feature matching

The relative camera motion between a set of images can be determined with the use of corresponding features. A standard exhaustive matching approach will attempt to match every image against every other image. Since in this approach the number of matching candidates increases quadratically with the image count, exhaustive matching is only viable with a relatively low number of images. For larger datasets Colmap employs a sequential approach for ordered images sets with consecutively captured images. For unordered datasets a vocabulary tree approach can be used. A vocabulary feature descriptor tree was trained from previous reconstructions and subsequently visual nearest neighbors can be determined for new images. Matching can then be performed on these nearest neighbors [20]. A final approach, found in all 3 packages, is spatial matching where spacial nearest neighbors are determined using prior information, such as GPS coordinates in the EXIF data. Neither commercial packages discloses internal matching approaches. A *generic preselection* mode, available in Photoscan, determines overlapping pairs of photos by matching them using lower accuracy setting first, however details of this method are missing.

2.3.4 Camera pose estimation

The extracted image correspondences are used to estimate camera poses, camera internal parameters and 3D coordinates of image points. Two major pipelines can be distinguished to perform this step: incremental and global. An *incremental* pipeline is the standard approach that adds one image at a time, calculates the unknown parameters and thus grows the reconstruction. Due to a potential buildup of error, better known as drift, in this process it requires repeated operations of bundle adjustment (BA) [24]. This heavily impacts performance for large datasets. A *global* reconstruction pipeline is different since it considers an entire view graph at the same time instead of incrementally adding images to a reconstruction [12]. This way only a single iteration of the BA is required. While much more efficient, it may be more sensitive to outliers.To

tackle the issues of efficiency, accuracy and robustness, recent efforts focus on the implementation of a hybrid reconstruction technique [6].
Our Colmap-based implementation employs an incremental reconstruction pipeline. Agisoft Photoscan likely also implements an incremental SfM pipeline. Evidence for this is the similar performance with regards to Colmap as well as the repeated BA calls visible in the logs. RealityCapture likely uses a global pipeline or slight variant.

2.3.5 Dense reconstruction and model generation

Once a sparse representation of the scene has been completed, denser scene geometry may be recovered. Typical dense reconstruction pipelines produce depth maps from stereo-pairs for all registered images. This relies on accurate exterior and interior camera parameters and epipolar geometry between images to constraint the search for matches [18]. Other methods include the use of region growing [22] or graph-cuts [14]. Depth maps are subsequently fused into a dense point cloud. Finally a dense surface is estimated from this fused point cloud typically using Poisson surface reconstruction methods [13].

3 Experiments

For consistency purposes, the processing of the datasets was done on the same computer. Following versions and general settings of the software were used:

- Capturing Reality RealityCapture 1.0.3.4658
  Standard 'medium' settings were used.
- Agisoft Photoscan 1.4.0
  Standard 'High' accuracy settings were chosen, lower values would cause internal image downsizing. A Generic preselection (see 2.3.3 matching approach was enabled
- Colmap 3.4 based implementation
  A custom implementation was deployed. Used photogrammetry approaches differ for the datasets and will be explained hereafter.

3.1 Dataset 1 - The Schiller Monument (464 images)

The first dataset depicts an historical statue in the center of the *Gendarmenmarkt* square: the *Schiller Denkmal*. This dataset was extracted from a single video. A total of 464 frames were extracted using $N_f = 5$. It is characterized by its sequential frame layout, very high overlap and loops.

Colmap matching strategy Due to the sequential ordering, optimizations in the matching part of the reconstruction pipeline could be employed. Colmap implements a sequential matching approach to optimize the pipeline performance. Here, each available frame will be matched with the next N consecutive frames. Tests have shown that an slightly adapted approach to this method yields more consistent results and reduces drift. In this approach each frame f_i is matched with $f_{i+1}, f_{i+4}, f_{i+9}, f_{i+16}, \ldots$. Build-in loop closure detection is available in Colmap's sequential matching strategy. For this a vocabulary tree approach is used as explained in section 2.3.3.

Figure 3 depicts the reconstruction in all packages. The results of the sparse reconstruction was similar in the three cases and the dense matching was performed as well. However, there was an issue with the Colmap-based implementation of the 3D textured model generation, which failed due to the high number of points, computed by the dense matching. This means we could send two models to the VR game designer and it was concluded that models from Realitycapture were preferred for further processing, due to their higher quality appearance.

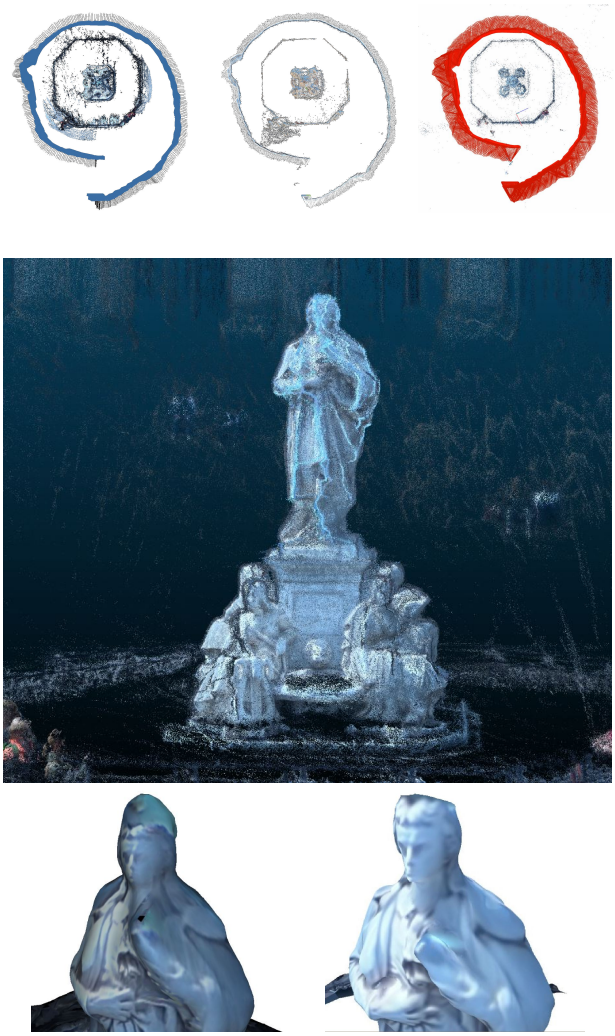


Figure 3: top: Sparse reconstruction in all 3 software packages. from left to right: Photoscan, Realitycapture, Colmap. Middle: dense point cloud from Colmap. Bottom: close-up of resulting model from Photoscan (left) and Realitycapture (right)

3.2 Dataset 2 - The Berlin *Gendarmenmarkt* square (1057 images)

This dataset contains partially ordered imagery from the entire *Gendarmenmarkt* square. The dataset consists of 1057 images with the following content distribution: 887 frames from 11 different video shots, 163 frames from publicly on-line available videos and 8 separate publicly on-line images. Prior knowledge of focal length was present in 2 images, for all other data no prior camera knowledge was available.

A first reconstruction using Photoscan (*'Photoscan_1'* in table 1) was heavily misaligned as shown in Figure 4 (b). A potential reason for this might be Photoscan's assignment of shared camera intrinsics between all same-size sensors. Due to high optical difference in the various lenses, a bad camera calibration and thus bad sparse alignment with an average reprojection error of 16px, occurred. In a second try, a successful (*'Photoscan_2'*) reconstruction was obtained using only the 887 frames.

Colmap's vocabulary tree matching alogrithm followed by its incremental pipeline managed to succesfully align 1032 images. The resulting sparse reconstruction consisted of 176130 tiepoints,

Table 1: Overview of the results and performance of the different photogrammetric reconstruction methods of the total square. The failed reconstruction is shown in red.

	total images matched	images matched new frames / existing frames / existing img	avg reproj. error (px)	Tiepoints	Feature detection time (s)	Matching time (s)	Sparse reconstr. time (s)	BA time (s) (#iterations)	Model generation time (s)
Colmap	1032	861/163/8	0.54	176130	76	5405	4455	3843 (56)	N/A
Reality Capture	906	813/93/0	0.88	673757	1449 (combined)			undisclosed	1118
Photoscan_1	1036	864/163/8	16	336711	61	3543	805	554 (350)	N/A
Photoscan_2	879	879/0/0	0,3	337619	37	2588	700	532 (289)	752

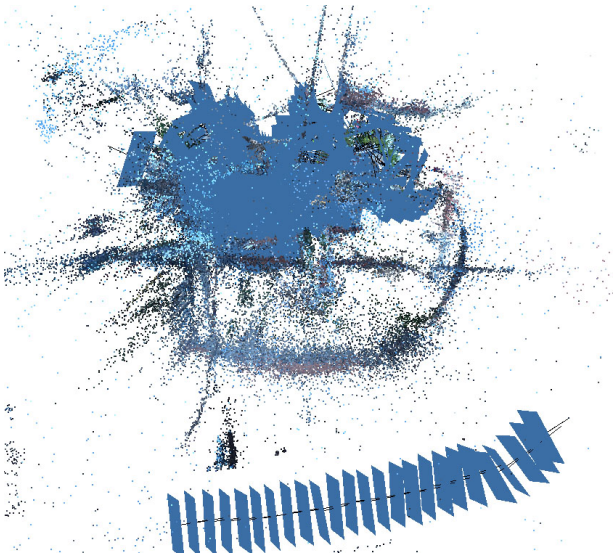


Figure 4: Incorrect image alignment in Photoscan using all available data

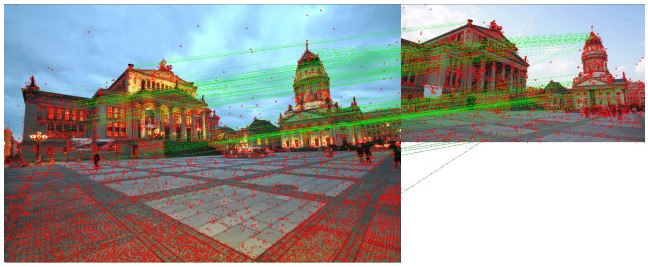


Figure 5: Evening image of the square (including exterior building lighting) matched to daytime image

remarkably lower than results from Photoscan and Realitycapture. This was due to the chosen thresholds of the triangulation process, such as a minimum ray intersecting angle of 3° and maximum reprojection error of 4px. The resulting sparse pointcloud from the colmap implementation is therefore cleaner with less visual artifacts (see Figure 6).

3.3 OpenStreetMap comparison

Since no ground-truth measurements such as GPS-points or feature distances (e.g. building width) are available for our reconstruction of the *Gendarmenmarkt*, a swift accuracy assessment was done using OpenStreetMap (OSM) data [16]. OpenStreetMap is a map of the world, created by users all over the world whose data and free to access. Figure 7 shows an orthographic top-view of our sparse point cloud projected on available *Gendarmenmarkt* GIS data. Structure edges from the *Konzerthaus* (middle left) were fitted onto the corresponding corners in the OSM data. The point cloud clearly coincides well with the facades and corners of the three buildings on the opposite side of the square. The fit with the churches on either side of the *Konzerthaus* seems worse but this is mainly due to the fact that OSM shows the total footprint of each building, which does not coincide with the main facades or walls.

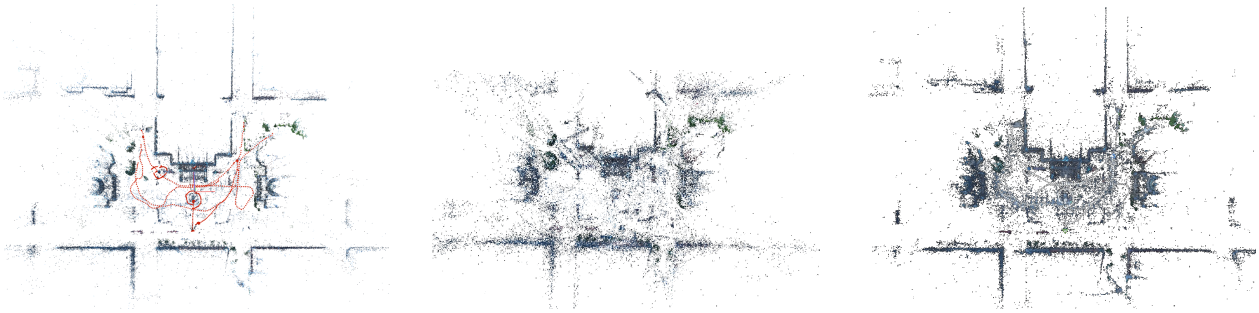
4 Virtual Reality Integration

The VR environment of the square was developed using the Unity3D game Engine [25]. The assets were imported into Autodesk Maya [3]. Prior to the editing in Maya, the environment was developed by placing the 3D models of the square in the scene along with smaller objects extracted from the market. Figure 8 shows the 3D environment in Unity with the extracted models. Once the 3D scene was developed, testing was done based on the hardware capabilities to produce an environment that can be supported by current devices.

The preparation of the model was done in Maya. After 3D extraction, minor retouching had to be done, as well as deletion of extra polygons in the model to make it usable in the VR environment due to hardware limitations on rendering too many polygons. It was also necessary in some cases to use tools and plug-ins for Maya to decrease the polygon count of extracted 3D models.

4.1 Change in design process

In the current design process, the designer has to use photos, available on the Internet to create an estimated building model and use self-developed rectified textures to produce a similar looking 3D model with estimated sizes and dimensions. With the 3D models, extracted by photogrammetry, the relative sizes of the buildings, more details on the dimensions of the building and other 3D models are available to the designer. Figure 9 represents the difference between the Concert house on the *Gendarmenmarkt* square designed by a 3D designer (left) and a 3D reconstructed model (right). The



(a) Sparse reconstruction Colmap (b) Sparse reconstruction Photoscan(_2) (c) Sparse reconstruction Realitycapture

Figure 6: Results of the sparse reconstruction of the total square for the different photogrammetric packages

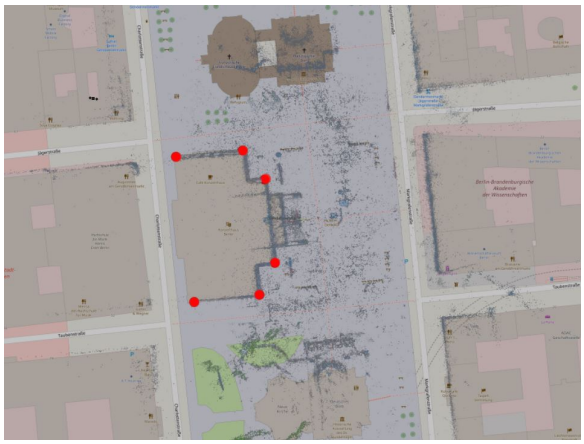


Figure 7: Point cloud data from the *Konzerthaus* (central building) was manually fit on Openstreetmap data (fitting points shown in red). The fit with the facades on the opposite side of the square is remarkably good.

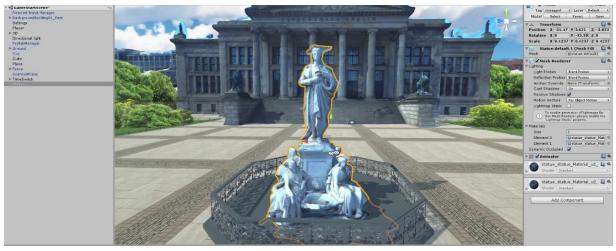


Figure 8: Extracted 3D model of Schiller Monument in Unity game engine

dimensions in the figure on the right correspond better to the real building which also helps the designer to modify their developed assets to the right dimensions.

Figure 9: Comparison between manually a modeled concert house (left) and the result from 3D reconstruction (right) in the VR environment. There is a clear difference in length proportions.

5 Conclusion and Future Work

This paper presents a novel workflow for game designers to optimize their current workflow of realistically digitizing existing sceneries and real world objects. The availability of 3D reconstructions, even if incomplete, has shown great potential for vr game development. Existing structures represent the reality more accurately when designing them.

5.1 Image processing pipeline improvements

The used 3D reconstruction pipeline proved to be successful. However, for several steps further improvements could be done.

Starting with the frame extraction where, using ffmpeg's *scene score*, shots are separated. Using a manual set threshold may cause issues for data with rapid camera movement or sudden light changes. Additional works like [8] improve this method by proposing an automated dynamic threshold model.

A second weakness in this pipeline is the constant frame rate extraction of each shot. Factors like frame overlap, camera movement or motion blur are not evaluated during extraction. While all three of these have a very large impact on reconstruction success rate and accuracy. For newly self-shot video optimizations can be done with the use of inertial motion unit (IMU) data. However, the focus of this paper rests on the reuse of existing data, therefor other solutions are presented. Prior research towards the selection of key frames for structure and motion recovery has been done before [2, 17]. Here a new keyframe is selected once the epipolar geometry model explains the relationship between a pair of frames better than the homography model. The distinction between both is based on the geometric robust information criterion (GRIC) values (*H-GRIC* and *F-GRIC*) [23].

Sudden peaks of motion blur during consecutive frames may be detected using Haar wavelet transform [11] by selecting the least blurred frame within a small range of consecutive frames. Thus minimizing the negative effects of sharp and sudden camera movement.
A clear difference in performance was noticed between the global reconstruction pipeline and incremental pipeline. The use of a Hybrid SfM pipeline could be evaluated for increased performance while maintaining accuracy from incremental reconstruction [6]. This hybrid pipeline was already successful on a dataset of the *Gendarmenmarkt* [6]. Furthermore it suits our dataset very well, since our data consists of several video sequences which may be aligned using a global reconstruction pipeline. Subsequently, frames withing each shot can be matched using an incremental SfM pipeline.

5.2 Dedicated online

The presence of a dedicated service which reuses existing content eliminates the need to shoot new material. This way vr development time gets reduced significantly.

REFERENCES

[1] Agisoft. Agisoft photoscan. http://www.agisoft.com, 2018.

[2] M. T. Ahmed, M. N. Dailey, J. L. Landabaso, and N. Herrero. Robust key frame extraction for 3D reconstruction from video streams. *Proceedings of the Fifth International Conference on Computer Vision Theory and Applications (VISAPP 2010)*, pp. 231–236, 2010.

[3] Autodesk. Maya. http://www.autodesk.eu/products/maya/overview, 2018.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. doi: 10.1016/j.cviu.2007.09.014

[5] CapturingReality. Realitycapture. http://www.capturingreality.com, 2018.

[6] H. Cui, X. Gao, S. Shen, Z. Hu, and C. Academy. HSfM : Hybrid Structure-from-Motion. In *CVPR*, pp. 1212–1221, 2017. doi: 10.1109/CVPR.2017.257

[7] Z. Cui and P. Tan. Global structure-from-motion by similarity averaging. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:864–872, 2015. doi: 10.1109/ICCV.2015.105

[8] A. Dimou, O. Nemethova, and M. Rupp. Scene Change Detection for H . 264 Using Dynamic Threshold Techniques. *Eurasip*, January 2014, 2005.

[9] Europeana.eu. Europeana collections, 2018. [Online; accessed 6-August-2018].

[10] FFMpeg-Developers. Ffmpeg 4.0.2. http://www.ffmpeg.org, 2018.

[11] Hanghang Tong, Mingjing Li, Hongjiang Zhang, and Changshui Zhang. Blur detection for digital images using wavelet transform. *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, 1:17–20, 2004. doi: 10.1109/ICME.2004.1394114

[12] N. Jiang, Z. Cui, and P. Tan. A global linear method for camera pose registration. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 481–488, 2013. doi: 10.1109/ICCV.2013.66

[13] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson Surface Reconstruction. *Proceedings of the Symposium on Geometry Processing*, pp. 61–70, 2006. doi: 10.1145/1364901.1364904

[14] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. *Computer Vision—ECCV 2002*, pp. 8–40, 2002. doi: 10.1007/3-540-47977-5_6

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. doi: 10.1023/B:VISI.0000029664.99615.94

[16] OpenStreetMap-contributors. OpenStreetMap (OSM). https://www.openstreetmap.org, 2018.

[17] M. Pollefeys, L. V. Gool, M. Vergauwen, K. Cornelis, F. Verbiest, and J. Tops. Video-to-3D. *International Archives of Photogrammetry, Remote Sensing and Spatial Information*, 34:252–257, 2002.

[18] F. Remondino, E. Nocerino, I. Toschi, and F. Menna. A critical review of automated photogrammetric processing of large datasets. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(2W5):591–599, 2017. doi: 10.5194/isprs-archives-XLII-2-W5-591-2017

[19] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision (ACCV)*, 2016.

[21] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[22] R. Sensing, S. I. Sciences, C. V. Symposium, D. Shin, J.-p. Muller, I. Group, M. Space, C. Physics, and D. Surrey. an Explicit Growth Model of. *ISPRS Technical Commission V Symposium*, XXXVIII, 2010.

[23] P. H. S. Torr, a. W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion estimation from uncalibrated motion sequences. *Int. J. of Comp. Vision*, 32(1):27–44, 1999. doi: 10.1023/A:1008140928553

[24] B. Triggs, P. Mclauchlan, R. Hartley, A. Fitzgibbon, B. Triggs, P. Mclauchlan, R. Hartley, A. Fitzgibbon, B. Adjustment, A. M. Synthesis, B. Triggs, A. Zisserman, and R. S. International. *Bundle Adjustment – A Modern Synthesis To cite this version : Bundle Adjustment — A Modern Synthesis*. Springer, 2010.

[25] Unity-developers. Unity3d. http://www.unity3d.com, 2018.

[26] K. Wilson and N. Snavely. Robust Global Translations with 1DSfM. pp. 61–75, 2014.