Markerless vision-based Augmented Reality for enhanced project visualization

Frédéric Bosché [1]*, David Tingdahl [2], Ludovico Carozza [3] and Luc Van Gool [2,4]

[1] School of the Built Environment, Heriot-Watt University, Edinburgh, UK
[2] ESAT/IBBT/VISICS, KU Leuven, Belgium
[3] ARCES, University of Bologna, Italy
[4] Computer Vision Lab, ETH Zurich, Switzerland
** Corresponding author (f.n.bosche@hw.ac.uk)*

Purpose This work aims to develop a system that enables improved visualization of Architectural Engineering and Construction (AEC) 3D-information for application in design, construction, and management of the built environment. **Method** A novel Augmented Reality (AR) system is presented that uses a single standard digital camera and which, contrary to other investigated approaches, does not rely on any markers inserted in the scene, nor on any positioning and inertial technologies. The system is solely image-based and consists of two stages. In a first offline stage, a 3D-map of the scene is automatically constructed from a set of digital images, and the augmenting information (e.g. the 3D-model of the building asset) is subsequently registered with this map. The 3D-map reconstruction employs structure-from-motion techniques with SURF features (the resulting map consisting of a set 3D-referenced SURF-features) followed by a Poisson mesh reconstruction procedure. The next step consists of online operations. The positions of target digital images (e.g. from video stream or head-mounted camera) are automatically calculated, using a robust SURF feature matching procedure that is optimized for three different situations (initialization, tracking, and resetting) implementing octrees for efficient 3D-pruning, and *k*d-trees for efficient feature matching. Once each input image is positioned within the map, the view is augmented. A notable feature of dense mesh scene reconstruction conducted in the present work is that it enables static occlusions of the scene on the augmenting data to be taken into account. **Results & Discussion** Several experiments validate the pr posed system and demonstrate its overall performance: a near real-time processing speed, very accurate and stable positioning. The limitations of the current system are also discussed including: the currently limited processing speed and the need for adequately textured scenes.

Keywords: *realities or application systems, augmented reality, image-based, markerless*

INTRODUCTION

Visualization in Construction

Construction projects are complex endeavors requiring the collaborative work of numerous different stakeholders, and generating large amounts of data and information from which complex decisions are made. Thanks to exponentially increasing computational capabilities, *Building Information Modeling (BIM)* is now being intensively developed with the aim of more efficiently and effectively managing lifecycle construction information. Rooted from 3D modeling and visualizations, BIM engines offer enhanced visualization and management of construction information. *Virtual Reality (VR)* immersive environments are further proposed in order to enhance user experience in navigating the created virtual worlds[1]. Numerous works have been published with regards to the development of VR environments[2,3,4]. Despite the great advances already made in developing and promoting VR in the Architectural, Engineering and Construction (AEC) industry, VR presents a couple of inherent limitations:

1. *Virtuality*: most developed VR technologies for the AEC industry focus on providing means to explore digital information in entirely digital worlds. As a result, VR is most useful during pre-project and design phases of construction projects, but is not fully adapted for construction and operation stages, where the virtual information may need to be more closely linked and visualized with the real world. It is noted though that some technologies are being developed to capture the state of actual construction projects and integrate it within the project Building Information Model[5,6]; but in these systems the visualization remains entirely within a virtual world.
2. *Single-user*: Many VR environments, i.e. VR Immersive rooms, focus on single user experience (only one “view” of the model can be seen at a time), preventing multiple users to simultaneously have their own views of the information. Nonetheless, we note that some more complex systems are being developed that enable multiple views simultateneously[7].

Augmented Reality

On the other end of the *virtual continuum* is *Aug-*

mented Reality (AR). AR aims at fusing virtual and actual information, e.g. by projecting virtual information on head-mounted displays (HMDs) that simultaneously enable the visualization of the real environment around. As a result, AR inherently has the potential to overcome the two main limitations identified above.

AR has already been investigated for application within the AEC industry with systems such as ARVISCOPE[8] and AR4BC[9]. These systems demonstrate the great potential identified above, but they also exemplify the challenges faced in developing such systems:

1. *Positioning*: In order to ensure a realistic and accurate overlaying of the virtual information on the viewed real world, the positions with regard to the real world of both the virtual information and the person (e.g. HMDs) must be known very accurately. In AR, small errors in those estimations rapidly result in obvious errors in the overlay. Regarding the AR systems reviewed above, it appears that, although systems based on positioning technologies like GPS and inertial sensors have the advantage not to require any prior knowledge of the scenes, they are fairly unstable due to the sensors inaccuracies.
2. *Occlusions*: Unless 3D information about the actual real world is available, occlusions of virtual objects by real-world objects are often not taken into account, resulting in obvious artifacts (ARVISCOPE[8] suggests the use of range camera to compute such occlusions in real-time, but these cameras only work for ranges lower than 10m).

This paper presents a novel AR system that is based on different technologies as those traditionally investigated. The main particularity of the system is that it does not rely on any beacon-based localization (e.g. GPS) or inertial navigation systems (although they could all be used complementarily). The system is solely image-based. This is achieved at the cost of a prior visit of the site of interest where numerous digital pictures must be acquired. The images are used by the positioning algorithm, but have the secondary advantage that they can be used to reconstruct a 3D model of the site, that can be used to compute static occlusions of the site on the augmenting information, an advantage over previous approaches.

SYSTEM OVERVIEW

The system is composed of two stages. In an *off-line mapping* stage, the actual 3D scene is first learnt and then augmented with virtual elements. Subsequently during *on-line* operations, for each image of the input stream (*target images*), the camera pose is first estimated, and the image is then augmented with appropriately occluded virtual scene objects.

These two stages are detailed in the following two sections. Then, validation experiments are presented that demonstrate the performance of the system.

OFF-LINE MAPPING

The offline mapping process is composed of two sub-stages detailed below: (1) learning the scene; (2) augmenting the scene.

Learning the scene

The input to the learning stage includes a set of images of the scene of interest, called *training images*, with corresponding camera intrinsic parameters. The mapping process, summarized in Fig.1, is fully automated and goes as follows. First, *Speeded Up Robust Features* (*SURF)*[10] are extracted from all training images. These SURF features are used in a *Structure-from-Motion (SfM)* framework to recover the scene 3D structure. SURF features are used in an initial sparse matching step to select candidate image triplets for projective reconstruction. The robustness of SURF features to scale changes allows some constraints about camera motion to be relaxed (normally constrained to turn around the building to be reconstructed), permitting camera paths at different distances from the building. A subsequent robust Euclidean *Bundle Adjustment* from candidate views directly registers the 2D SURF descriptors in the reconstructed Euclidean 3D reference frame to build the map of 3D-referenced features. This approach effectively populates the map with 3D-referenced SURF features. We use the ARC3D framework[11] for 3D scene mapping and self-calibration.

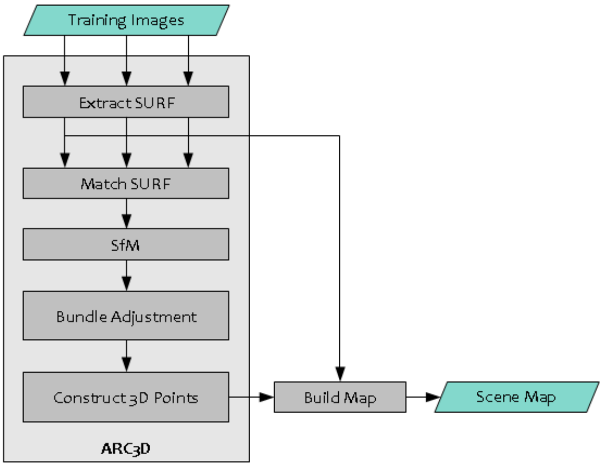

Fig.1. Off-line mapping process

Augmenting the scene

ARC3D actually provides us with an additional feature that is of particular interest to our system. In addition to learning a 3D map of SURF features, ARC3D enables a dense reconstruction of the acquired scene, in the form of a *3D (textured) mesh*, using the same input images (we use Poisson mesh reconstruction for this). Compared to the point cloud of the reconstructed map, this mesh presents two advantages:

1. It simplifies the manual insertion of virtual objects in the scene (discussed below).
2. During online processing, it enables the computation of occlusions by the scene of the virtual objects and vice versa.

Given the dense 3D mesh of the scene, the user can easily insert virtual (augmenting) 3D objects within the scene. Note that, in the case when a virtual object is planned to replace an existing one (e.g. a building is planned to be demolished and replaced by a new one), the user just has to remove from the reconstructed mesh the parts corresponding to the objects to be replaced. This ensures that occlusions caused by the objects to be replaced are not taken into account when augmenting the target images with the new objects. Fig. 4 in section Validation Experiments shows an example of a reconstructed scene augmented with a virtual building.

ON-LINE IMAGE STREAM PROCESSING

Image Positioning

During on-line operations, the system processes the *target image sequence* (e.g. from a video stream). For each target image, SURF features are extracted and the S_{target} (=1500) strongest ones are matched with the SURF descriptors in the database (using the Euclidean distance in a 64-dimensional space). Matched feature descriptors permit to establish correspondences, called *matched 3D points*, between the 2D image coordinates of the target image features and the 3D coordinates associated to the matched map features. Knowing the target camera intrinsic parameters, the camera pose is then estimated from these correspondences by wrapping the 3-point algorithm[12] in a *Random Sampling And Consensus (RANSAC)* framework[13] – i.e. triplets of feature matches are iteratively tested in 3D to find the one that leads to the most matches being geometrically correct. The resulting initial pose estimation is subsequently used in a *Guided Refinement* process, in which the database 3D points culled using the frustum from the initial pose estimate are reprojected on the image plane of the target image, and matches to the target image SURF features are identified only within a radius of ρ_{2D} pixels (in our implementation, $5 \leq \rho_{2D} \leq 15$ pixels). This process enables to reassess all initial matches and identify additional ones. Finally, a refined pose estimation is obtained by putting all matches into a Levenberg-Marquardt non-linear pose optimization algorithm[14].

Image Positioning Optimization

While the method above can enable robust pose estimations, it would require a brute force matching of the target image features with all features in the learnt scene map. In other words, its complexity is proportional to the size of the scene map, i.e. database of 3D-referenced SURF descriptors, which would prevent real-time applications. To avoid latency in the system, we implement different techniques to expedite matching without jeopardizing the quality of the pose estimates. These are presented in the following subsections.

K-d tree of scene database features

We use the common strategy consisting of partitioning the SURF descriptor space into a *k-d tree*[15], so that, when matching each target image feature descriptor, only the subspace associated with the target descriptor is visited. This effectively reduces the computational payload.

Filtering scene database features by strength

SURF features can be given a value of reliability, or *strength*, which is associated to the Hessian response[10]. Strength depends on the scene feature, the viewpoint and the lighting condition of the image, thus capturing the repeatability of the feature.

However, this repeatability measure is specific to the image from which each feature is extracted, while each scene database 3D points is calculated from the matching of features extracted from different images. In order to obtain a *global SURF strength*, we thus propose to assign to each 3D reconstructed point the average of the strengths of the SURF features corresponding to that point in the different input images. This way, SURF descriptors can be globally sorted and only those with a high average repeatability can be retained. Of course, more sophisticated weighting algorithms of the different strengths could be implemented.

Multiple Matching

Due to the repetitions and self-similarities often observed in urban architecture, there is a high likelihood that any target image feature be matched with high confidence with several database features. Enabling matching with the best matched feature only would create the risk of wrong matches and consequently wrong pose estimations (mathematically right, but actually wrong). It is thus proposed to enable one-to-many matches between each target image feature and the database features. Correct poses are then identified through the RANSAC-based pose estimation framework.

Additional heuristics are proposed to increase matching performance depending on the three configurations, or *modes*, that can be encountered during the processing of the stream of target images: (1) Pose initialization; (2) Pose tracking; and (3) Pose resetting. These three modes are further described below. Fig. 2 summarizes the strategy used during on-line processing to position each target image within the learnt 3D map.

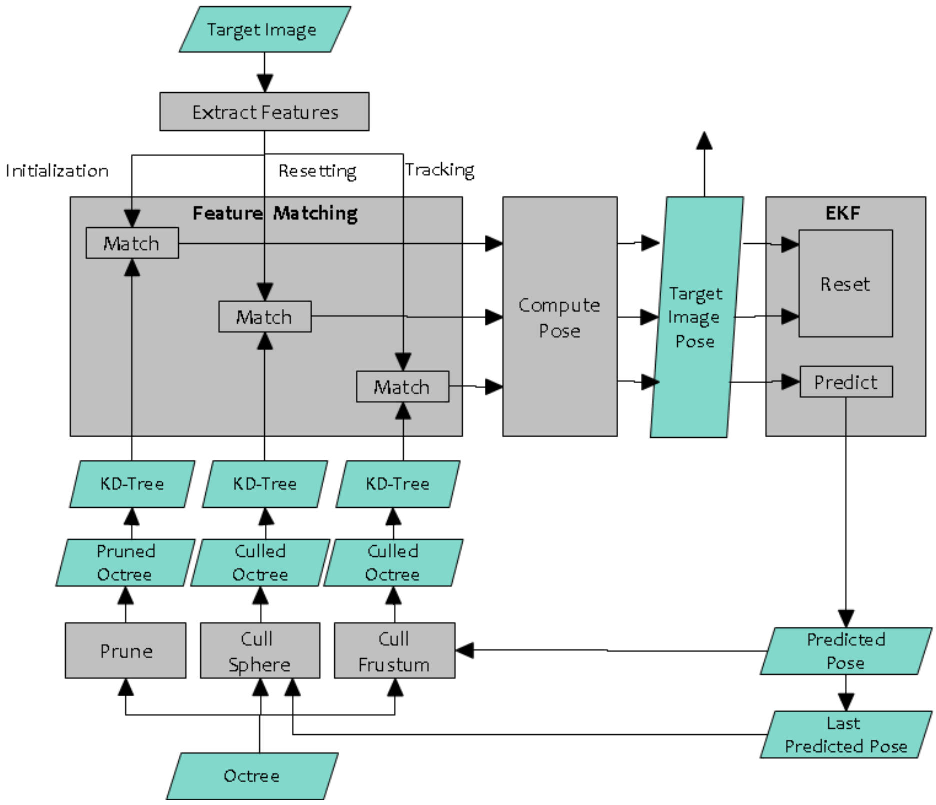

Fig.2. On-line image positioning process, emphasizing the strategies chosen to reduce the search space for efficient feature matching

Pose initialization

Pose initialization is the mode when no prior knowledge about the pose of the camera is available, e.g. at the beginning of the processing of the image stream, or when tracking has failed for n consecutive images (we use n=20).

In this situation, matching must be made considering a set of database features well spread within the entire scene. To achieve this, we arrange the 3D-referenced feature points into an *octree*, where each node represents a partition (cuboid) of the 3D scene. The octree is populated with all scene 3D points, splitting each node once the number of points it contains reaches a threshold N'_{max}.

While the full octree is used in the other two modes described below, during pose initialization, a *pruned octree* is used to speed up matching. The pruned octree is constructed by removing all octree cells with a volume smaller V_{min}. The points within those cells are combined in the parent cell (with volume larger than V_{min}) and only the N_{max} points with the largest *global SURF strengths* are retained. The resulting pruned octree contains much fewer points that the entire one, but these cover the entire scene as homogeneously as possible. In our implementation, we use N_{max} = 200 and N'_{max} = 400 and V_{min} = 5m^3. Note that the pruned octree is only computed once offline.

Pose tracking

In this mode, some knowledge about previous camera poses is available. Assuming linear camera dynamics, a prediction of the pose of the current camera is made using an *Extended Kalman Filter (EKF)*[16]. The frustum of the predicted camera pose is then used to cull the full octree (near and far culling planes are used with distances set to 0m and 50m respectively). Furthermore, only the $S_{frustum}$ strongest features of the points located in the predicted view are considered for matching and are organized in a k-d tree. In the experiments presented below, we use $S_{frustum}$=S_{target}.

To prevent the system from considering unreasonable pose predictions, we reject any prediction with a change in camera orientation larger than $\delta\alpha_{max}$=0.1rad (≈5deg), in which case the camera pose is computed in Reset mode (see below).

Pose resetting

In this mode, tracking has failed for the given image, but was successful for at least one of the last n target images (see Initialization mode). Given the location of the last successfully calculated pose, we then cull the full octree using a sphere centered at that location and with a radius of ρ_{sphere}=50m. Additionally, like in Tracking mode, only the S_{sphere} points with the strongest *global SURF strengths* are kept for matching and are organized in a k-d tree. In the experiments presented below, we use S_{sphere}=4$S_{frustum}$.

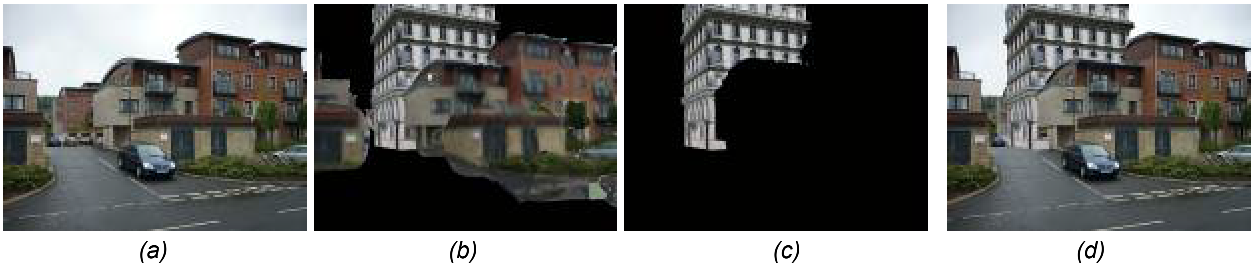

Fig.3. The process to augment a target image: (a) Target image; (b) Camera positioned in the reconstructed and augmented virtual environment; (c) Texture to be projected on the target image; (d) Augmented image

Augmenting Stage

Once the system has confidently calculated the pose of the camera corresponding to a target image, this image is augmented. In order for this augmentation to take occlusions from the reconstructed 3D scene into account, the simple procedure shown in Fig. 3 is used, that simply aims to reproject on the target images the parts of the augmenting object(s) that are not occluded by the reconstructed 3D scene mesh.

VALIDATION EXPERIMENTS

The proposed AR system has been tested using several different urban scenes, with different levels of complexity with regard to the amount of texture, as well as the repetition of textures (which can confuse the system). Two experiments are presented here. The first is detailed, and aims to highlight the overall performance of the system. The second experiment demonstrates the performance of the system in a very different context. The attached video illustrates (1) the detailed stages and results achieved in the first experiment, and (2) the results achieved in the second experiment.

Experiment 1: Housing Estate in Edinburgh

This experiment was conducted in a modern housing estate composed of apartment buildings and located in Edinburgh, Scotland. 22 training images of a part of the estate were taken by a person walking around it. The pictures have a 2048x1536 resolution, i.e. ~3M pixels (see attached video). The processing of the images results in two files, containing the list of 3D-referenced SURF features and a 3D mesh of the scene, which is then augmented with an additional building (see Fig. 4). A set of target images was acquired later. The same digital camera was used with the same initial resolution (2048x1536). In order to simulate a video sequence, 120 images were acquired in 'burst mode'.

In order to assess the impact of image size on the system's performance (including when the training and target images have different sizes), the dataset above was duplicated with all images downsampled to 640x480.

Fig. 5 shows the results obtained for some of the input target images with resolution 2048x1536. A visual analysis of all the results (see attached video as well) for all target images shows that all poses were successfully calculated. Nonetheless, it is noted that at four occasions when a target image was processed in tracking mode, a sharp acceleration in the camera orientation occurred and the EKF prediction resulted in a change of camera orientation, $\delta\alpha$, slightly larger than $\delta\alpha_{max}$. This resulted in the estimated pose being rejected by the system and recalculated (successfully) in Reset mode. The reason for these rejections is that the low-frequency of the data acquisition in 'burst mode' (~1fps) made possible significant camera motions between frames. Even then, the system achieves accurate pose estimations and effectively recovers from tracking failures.

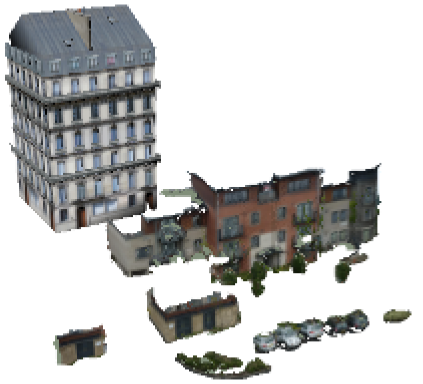

Fig.4. Exp. 1 - Augmented reconstructed 3D scene

Tab. 1, 2 and 3 present pose estimation performance results obtained using the dataset with all images having 2048x1536 resolutions. The three tables report results for experiments conducted with and without two options: *Multiple Matches (MM)* and *Guided Refinement (GR)*. Tab. 1 reports the average numbers of features used for matching and the average numbers of matches obtained (for the three possible modes). Tab. 2 reports the success rate in pose calculation, based on the software's own assessment criteria. Finally, Tab. 3 reports the average processing times obtained for the different modes. The analysis of these results shows that, as expected, MM and GR tend to improve the success rate in pose calculation. However, in this experiment at least, the improvement is not critical, since it does not significantly improve the pose estimation success rates (a visual analysis of the results shows that, in that particular experiment, the software actually achieves 100% in all cases.). Then, Tab. 3 shows that the use of GR significantly impacts the average processing time with an average increase of ~40%, and that the current implementation of the proposed tracking system does not enable processing speeds that would support real-time applications. The issue of processing speed is investigated further below.

Tab. 4 compares the performance achieved by the system for training and target images with different resolutions, after visually controlling the results. Clearly, the system performs best when the training and target images have a similar size. While this is generally not surprising, it also shows that the scale-invariance property of SURF features[10] can be put to the limit if the difference in image resolution is significant (given that the scene is observed from similar distances). These results also tend to show that small images actually achieve similar pose estimation performance as large images, with the advantage of faster processing times (see Tab. 5).

Tab. 5 presents computational times similarly to those in Tab. 3 but for training and target images with resolution 640x480. It appears that, although the number of image pixels is effectively reduced by a factor of 10, the computational efficiency does not improve that significantly. This is due to the fact that, although the computation of the SURF features for the target image is significantly sped up, the parameters S_{target}, $S_{frustum}$ and S_{sphere} remain unchanged, so that the system calculates a similar number of matches.

Tab. 6 presents computational times achieved with training and target images with resolution 640x480, and the parameters S_{target}, $S_{frustum}$ and S_{sphere} set to S_{target} = $S_{frustum}$ = 500 and S_{sphere} = 2, 000. The pose estimation quality are not impacted by these settings, but the processing times are further decreased (although not that significantly).

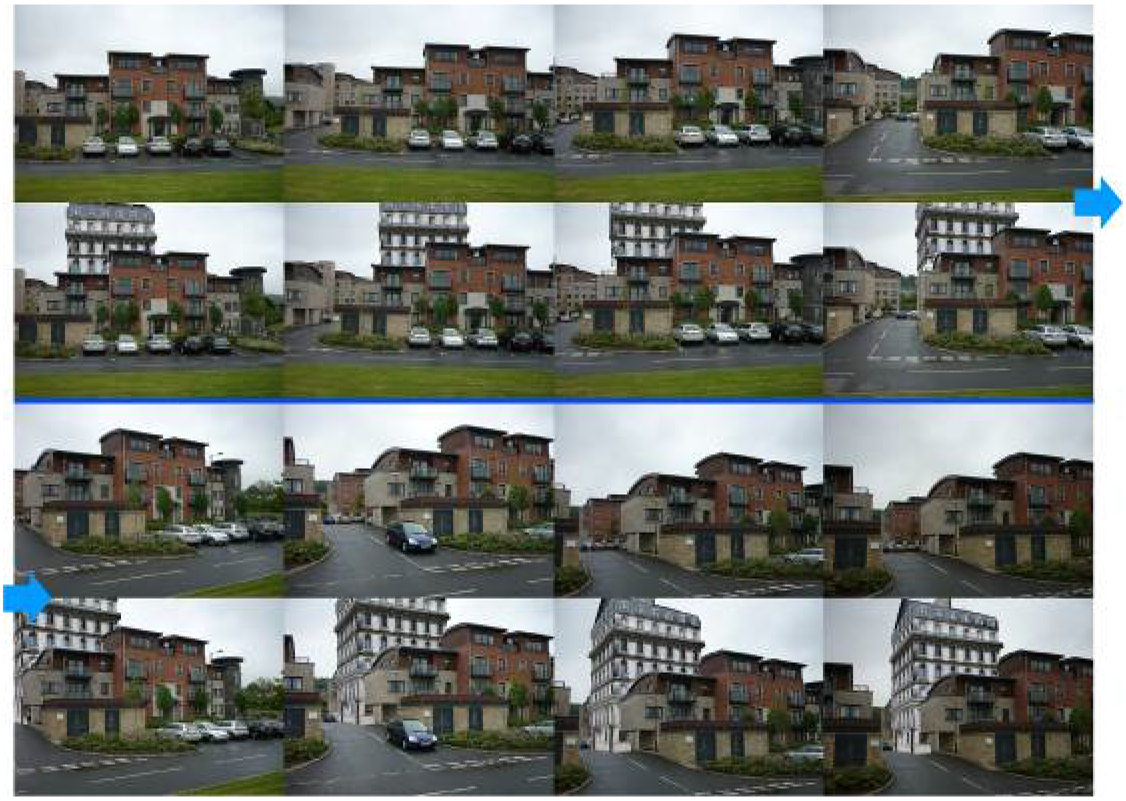

Fig.5. Experiment 1 - Eight of the 120 target image stream before (lines 1 and 3) and after being augmented (lines 2 and 4)

Tab. 1. Experiment 1 - Statistics of the pose calculation performance for the three modes Initialization, Tracking and Reset. The columns Matching and GR report the results obtained after the initial matching stage and the guided refinement stage respectively. The columns DB and Match report the number of map features used for matching and the number of matches found

MM	*GR*	*Initialization*				*Tracking*				*Reset*			
		Matching		*GR*		*Matching*		*GR*		*Matching*		*GR*	
		DB	*Match*	*DB*	*Match*	*DB*	*Match*	*DB*	*Match*	*DB*	*Match*	*DB*	*Match*
No	No	2,675	114	N/A	N/A	1,500	72	N/A	N/A	6,000	85	N/A	N/A
No	Yes	2,675	114	1,500	232	1,500	75	1,500	156	6,000	85	1,500	170
Yes	No	2,675	119	N/A	N/A	1,500	88	N/A	N/A	6,000	88	N/A	N/A
Yes	Yes	2,675	119	1,500	286	1,500	88	1,500	217	6,000	89	1,500	214

Tab. 2. Experiment 1 - Statistics of the pose calculation performance, as reported by the system. In brackets are the numbers of images processed in the particular mode.

MM	GR	Success rate (system)		
		Initial.	Tracking	Reset
No	No	100% (1)	95% (124)	100% (6)
No	Yes	100% (1)	97% (124)	100% (4)
Yes	No	100% (1)	97% (124)	100% (4)
Yes	Yes	100% (1)	98% (124)	100% (3)

Tab. 3. Experiment 1 - Average computation times for pose calculation using training and target images having all 2048x1536 resolution.

MM	GR	Mean processing time (s)		
		Initial.	Tracking	Reset
No	No	1.34	2.16	1.18
No	Yes	2.00	2.81	1.81
Yes	No	1.37	2.17	1.18
Yes	Yes	1.99	2.81	1.81

Tab. 4. Experiment 1 - Comparison of the pose estimation performance for different combinations of sizes of the training and target images. Small (S) images have 640x480 resolution, and large (L) images have 2048x1536 resolution.

Image size		MM	GR	Success rate (visual)		
Train.	Target			Initial.	Tracking	Reset
S	S	No	No	100%	100%	100%
		No	Yes	100%	100%	100%
		Yes	No	100%	100%	100%
		Yes	Yes	100%	100%	100%
S	L	No	No	100%	88%	45%
		No	Yes	100%	88%	54%
		Yes	No	100%	88%	70%
		Yes	Yes	100%	85%	75%
L	S	No	No	100%	98%	100%
		No	Yes	100%	99%	100%
		Yes	No	100%	94%	88%
		Yes	Yes	100%	94%	89%
L	L	No	No	100%	100%	100%
		No	Yes	100%	100%	100%
		Yes	No	100%	100%	100%
		Yes	Yes	100%	100%	100%

Tab. 5. Experiment 1 - Average computational times for pose calculation using training and target images having all 640x480 resolution.

MM	GR	Mean processing time (s)		
		Initial.	Tracking	Reset
No	No	0.58	0.75	0.75
No	Yes	1.25	1.22	1.21
Yes	No	0.55	0.57	0.56
Yes	Yes	1.20	1.20	1.20

Tab. 6. Experiment 1 - Average computational times for pose calculation using training and target images having all resolution 640x480, and the following parameter values are changed: S_{target} = $S_{frustum}$ = 500 and S_{sphere} = $4S_{frustum}$ = 2, 000.

MM	GR	Mean processing time (s)		
		Initial.	Tracking	Reset
No	No	0.35	0.26	0.34
No	Yes	0.45	0.39	0.46
Yes	No	0.34	0.28	0.34
Yes	Yes	0.45	0.40	0.47

Experiment 2: Seoul Imperial Palace

The results of this experiment are shown in Fig. 6 and on the video. The experiment used 80 training images (surrounding the temple) and 90 target images (acquired in camera 'burst mode'). All images have 2560x1920 resolutions. Fig. 6 and the video show the stability of the pose estimation algorithm of the system.

However, a limitation of the current process is illustrated in Fig. 7, which shows results for the same experiment, but with the augmenting virtual building positioned behind the temple. In that particular context the dense mesh reconstruction using the ARC3D framework didn't achieve sufficiently good quality with numerous holes in the final mesh, so that numerous artefacts appear when computing static occlusions.

Note that this issue could be addressed by using accurate 3D urban reconstructions (e.g. GIS level 2), align them with the ARC3D reconstructions and use them instead to calculate occlusions.

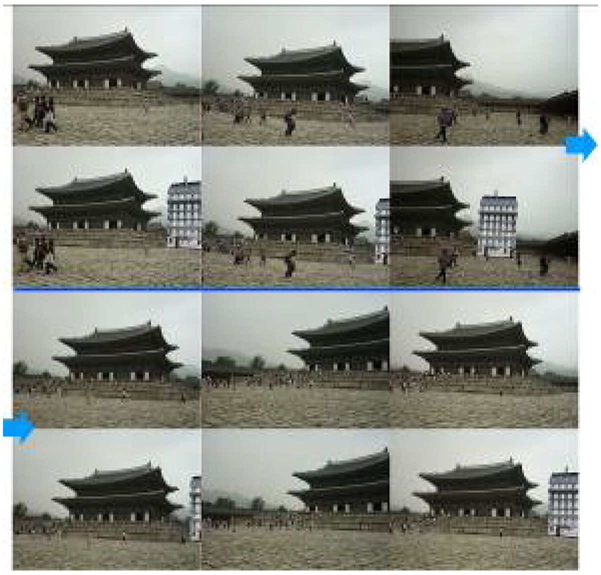

Fig.6. Experiment 2 – Six of the 90 target image stream before (lines 1 and 3) and after being augmented (lines 2 and 4).

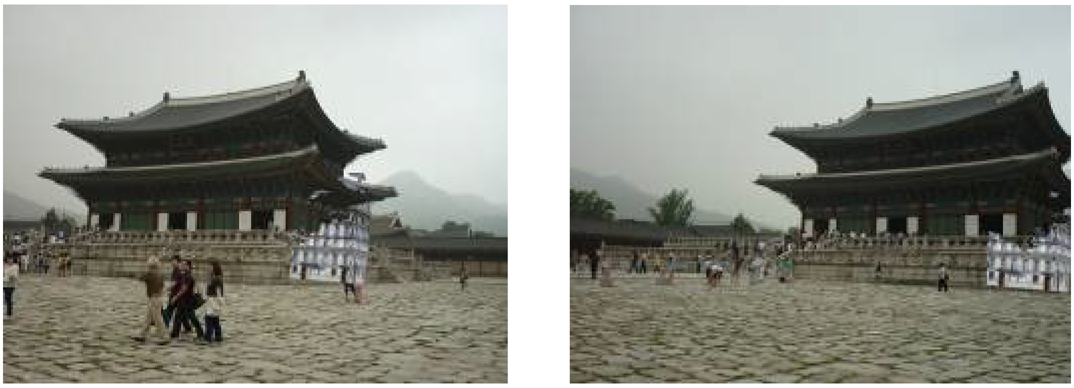

Fig.7. Experiment 2 – In this experiment, the Poisson reconstruction did not achieve good enough results for convincing occlusion calculations. These images illustrate how the bottom part of the existing building is not successfully reconstructed, resulting in missing occlusions of the actual scene on the inserted building

CONCLUSION

A markerless monocular vision-based augmented reality system has been presented, with the aim of providing AEC professionals with a tool enabling them to assess project 3D digital information (e.g. BIM) within their actual environment.

The performance of the system was successfully demonstrated on real imagery from two different scenes. The accuracy of the estimated poses was not directly estimated since no ground truth was available. However, the quality of the augmented images – in particular the calculation of occlusions – provides some clear observations of this accuracy.

Nonetheless, improvements could be made in several areas:

- The system in its current implementation only achieves up to 3fps which is not fast enough to consider a real-time AR system. While some improvement could be achieved by varying some parameters (e.g. S_{target}, $S_{frustum}$, S_{sphere}), transferring some data processing on the GPU also seems necessary. Nonetheless, as shown in this paper, the system may already be used in an "off-line" manner by augmenting a video input.
- Compared to other commonly used approaches, our tracking strategy does not rely on tracking image features (e.g. KLT[17]). Instead, it tracks the camera. While this approach may be more robust with respect to sharp changes in camera motion, the overall tracking is likely not as efficient. A combined system could be envisaged.
- Similarly, the current system does not rely on any global positioning or inertial system. While this brings some advantages, it also brings some limitations (e.g. GPS would be useful for initialization and resetting modes). A hybrid pose estimation module could thus be investigated.
- Since the focus of this work is on urban augmented reality, positioning techniques based on planar structures[18] could be investigated.
- Further culling of the database features may also be achieved by considering some feature visibility criterion[19].
- While the system is designed to handle scenes of the size of a neighbourhood, further testing needs to be conducted using larger reconstructed scenes.

Finally, it must be emphasized that there is one limitation that is inherent to vision-based localization approaches, which is that they perform adequately only when the scene presents sufficient structure. Therefore, the current system would likely fail in the case of greenfield projects with little built environment in the surroundings.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the Swiss National Science Foundation (SNSF) with the project Aerial Crowds #CRSI20 122696, as well as the European Space Agency (ESA) with the project Wear++. Ludovico Carozza would like to thank ARCES - University of Bologna for having supported his research period at ETH Zurich, through the "Marco Polo" scholarship.

References

1. Issa, R.R.A., "Virtual Reality: A solution to seamless technology integration in the AEC industry?", *ASCE Conference Proceedings*, Vol. 278, pp. 1011-1012, 2000.
2. Goulding, J., Nadim, N., Pedritis, P. and Alshawi, M., "Construction industry offsite production: A virtual reality interactive training environment prototype", *Advanced Engineering Informatics*, Vol. 26(1), pp. 103-116, 2012.
3. Austin, S. and Soetanto, R., "The use of ACT-UK Virtual Reality Simulation Centre to enhance the learning experience of undergraduate building students", *Engineering Education*, Vol. 5(1), pp. 2-10, 2011.
4. Bassanino, M., Wu, K.-C.,Yao, J., Khosrowshahi, F., Fernando, T. and Skjærbæk, J., "The Impact of Immersive Virtual Reality on Visualisation for a Design Review in Construction", *14th International*

Conference on Information Visualization (IV), pp. 585-589, 2010.

5. Turkan, Y., Bosché, F., Haas, C.T. and Haas, R., "Automated progress tracking using 4D schedule and 3D sensing technologies", *Automation in Construction*, Vol. 22, pp. 414-421, 2011.
6. Golparvar-Fard, M., Peña-Mora, F., and Savarese, S., “Automated model-based progress monitoring using unordered daily construction photographs and IFC as-planned models.” *ASCE Journal of Computing in Civil Engineering*, 2012.
7. Reynolds, M., Schoner, B., Richards, J., Dobson, K. and Gershenfeld, N., “An Immersive, Multi-User, Musical Stage Environment”, *SIGGRAPH*, 2001.
8. Behzadan, A.H. and Kamat, V., “Scalable algorithm for resolving incorrect occlusion in dynamic augmented reality engineering environments”, *Computer-Aided Civil and Infrastructure Engineering*, Vol. 25(1), pp. 3-19, 2010.
9. Woodward C., Hakkarainen M., “Mobile Mixed Reality System for Architectural and Construction Site Visualization”, in *Augmented Reality - Some Emerging Application Areas*, Andrew Yeh Ching Nee (ed.), 2011.
10. Bay, H., Tuytelaars, T. and Gool, L. van, “Surf: Speeded up robust features. Lecture Notes in Computer Science”, *Proceedings of the European Conference on Computer Vision (ECCV)*, Vol. 3951, pp. 404-417, 2006.
11. Vergauwen, M. and Gool, L. van, “Web-based 3d reconstruction service”, *Machine Vision and Applications*, Vol. 17, pp. 411-426, 2006.
12. Haralick, R.M., Lee, C.N., Ottenberg, K. and Nolle, M., “Review and analysis of the three point perspective pose estimation problem”, *International Journal of Computer Vision*, Vol. 13, pp. 331-356, 1994.
13. Fischler, M.A. and Bolles, R.C., “Random sample and consensus: A paradigm for model fitting with application to image analysis and automated cartography”, *Communications of the ACM*, Vol. 24, pp. 381-395, 1981.
14. Nocedal, J. and Wright, S.J., *Numerical Optimization, 2nd Edition*, Springer, 2006.
15. Gordon, I. and Lowe, D.G., “Scene modelling, recognition and tracking with invariant image features”, *Proceedings of the 3rd International Symposium on Mixed and Augmented Reality (ISMAR)*. pp. 110-119. Arlington, VA, USA, 2004.
16. Bishop, G. and Welch, G., “An Introduction to the Kalman Filter”, Course 8, *SIGGRAPH*, University of North Carolina at Chapel Hill, 2001.
17. Shi, J. and Tomasi, C., “Good features to track”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 593-600, 1994.
18. Simon, G., Fitzgibbon, A. and Zisserman, A., “Markerless tracking using planar structures in the scene”, *Proceedings of the International Symposium on Augmented Reality (ISAR)*, pp. 120-128, 2000.
19. Wolf, J., Burgard, W. and Burkhardt, H., “Robust vision-based localization by combining an image retrieval system with Monte Carlo localization”, *IEEE Transactions in Robotics*, Vol. 21(2), pp. 208-216, 2005.