Reconstructing Fukushima: A Case Study

Akihito Seki[1] Oliver J. Woodford[2] Satoshi Ito[1] Björn Stenger[2]
Makoto Hatakeyama[3] Junichi Shimamura[3]
[1]Corporate Research & Development Center, Toshiba
[2]Cambridge Research Laboratory, Toshiba Europe Limited
[3]Power Systems Company, Toshiba
akihito.seki@toshiba.co.jp

Abstract

We present the application of 3D reconstruction technology to the inspection and decommissioning work at the damaged Fukushima Daiichi nuclear power station in Japan. We discuss the challenges of this project, such as the difficult image capture conditions (including under water), required use of limited imaging hardware, and capture by personnel inexperienced in 3D reconstruction. We present an overview of the system developed for this project, a real-time reconstruction pipeline with robust camera pose estimation, low-latency probabilistic dense depth estimation and a novel descriptor for point cloud alignment—the Co-occurrence Histogram of Angle and Distance (CHAD). We discuss the modifications required to standard algorithms in order to perform reliably in such a scenario. As well as quantitative evaluations of these components on existing datasets, we show qualitative 3D reconstruction results of debris from the damaged plant and its spent fuel pool. Such results have enabled planning of the critical process of debris removal, without the harmful requirement of extensive human presence on site.

1. Introduction

Vision-based technology enables new inspection and planning applications in challenging environments, without the need for human presence, providing the potential to greatly improve rescue and recovery efforts in dangerous disaster areas. Decommissioning work at the Fukushima Daiichi nuclear power station in Japan, which was severely damaged in the aftermath of the 2011 Tohoku earthquake and tsunami, is one such project.

An ongoing stage of this project is the removal of debris such as steel beams, wires, and blocks of concrete, enabling the remaining radioactive material to be made safe. Computing a 3D reconstruction of the site as it stands, to which CAD models can then be fitted to the various elements, is a crucial task in this stage. It allows the planning of debris removal to be undertaken in a safe environment, away from the site, using CAD simulations.

In this work we demonstrate the application of vision-based 3D reconstruction technology to the Fukushima project. We discuss how this technology performs in real and challenging scenes, and the improvements to standard algorithms required for robust and reliable performance in such scenes.

Vision technology has recently been employed in a number of similar projects, such as reconstruction of infrastructure for civil engineering [4, 13, 25], temporal change detection in the aftermath of a natural disaster [21], and real-time 3D mapping for robots assisting in disaster response [16]. However, the technology applied is heavily dependent on the constraints of each particular scenario. We discuss the constraints of the Fukushima project below.

1.1. Project challenges

Before discussing the technology, it is important to understand the challenges faced in this particular project. High levels of radiation and structural instability make the site a hazardous environment for workers. Remotely controlled sensors therefore need to be used, but this is also challenging due to limited access. The debris to be removed is both above and below water, the latter in the spent fuel pools, so either a system that can work in both environments, or two separate systems, is required. Cranes have been erected on site, from which sensors can be hung, but wind and crane motion means that the sensors are often swinging, so the reconstruction pipeline needs to be robust to this. The radiation also affects sensors, therefore special radiation (and water) proof sensors must be used. The only such sensors in existence are encased video cameras, with a narrow field of view (approximately 30°). The actual sensor setup used, a radiation-proof video camera hung from a crane and moved about, is shown in Fig. 1. One final challenge is that, due

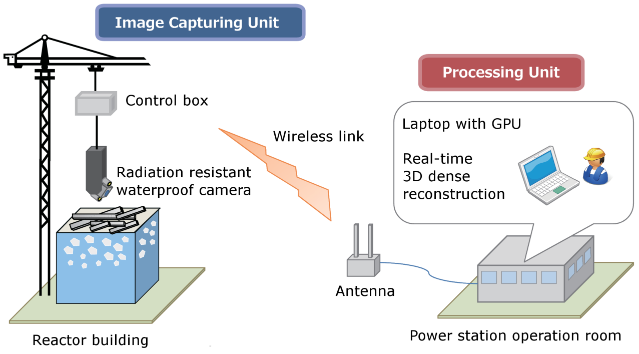


Image Capturing Unit
Control box
Radiation resistant waterproof camera
Reactor building
Wireless link
Antenna
Processing Unit
Laptop with GPU
Real-time 3D dense reconstruction
Power station operation room

Figure 1: **3D reconstruction system at the Fukushima Daiichi nuclear power station.** *Video captured with a radiation and water proof camera suspended from a crane is transmitted wirelessly to the operation room. The 3D structure is estimated in real-time, providing real-time feedback to the on-site workers controlling the crane.*

to the sensitivity of the project, no 3D reconstruction experts have been able to help in the image capture process. For this reason, it is important that the reconstruction system used provides real-time feedback to the team controlling the crane, so that they know either that the images captured can provide an adequate reconstruction, or that some further motion is required.

In the following section we describe our system, consisting of camera pose estimation, multi-view stereo, and relocalization. The method for registering 3D point clouds from multiple capture sessions is described in Section 3. We present quantitative evaluations of system components in Section 4, and qualitative results of the complete system in Section 5.

2. Our system

Our system consists of four main components: (1) real-time, monocular camera pose estimation based on a key frame SfM method, (2) real-time, monocular, video-based multi-view stereo, (3) real-time, image-based relocalization, and (4) registration of reconstructed point clouds using a novel 3D descriptor. We describe each of these components in the four subsections below, discussing earlier work relevant to each component in their respective subsection.

2.1. Camera pose estimation

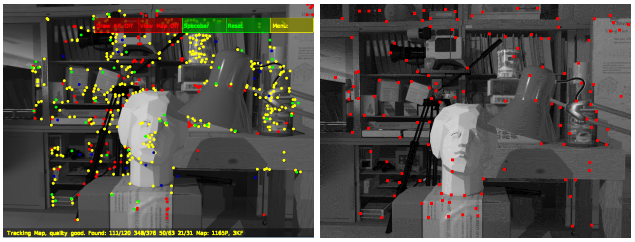
Camera pose estimation algorithms can be categorized into online and offline approaches. Online methods calibrate images one by one, as they arrive, *e.g.* [6, 10, 17], and generally make use of the temporal coherence of video by tracking feature points. Offline methods calibrate many images jointly in a single batch computation, *e.g.* [1, 24], and tend to assume that the images are unordered and taken from sparse viewpoints, hence match feature points using descriptors. In this work we are dealing with video input, and require real-time camera pose estimation in order to provide reconstruction feedback to the operator, therefore online methods are relevant. Three main categories of online approach exist: methods which track and triangulate sparse features in a probabilistic map, integrating information using a Kalman filter [6]; methods which track and triangulate sparse features in key frames using bundle adjustment, minimizing reprojection error [10]; finally, methods which construct a dense model of the scene and register frames to this by minimizing photometric error [17]. We use the second of these approaches, as it is robust to changes in lighting/exposure, in contrast to the last category, and provides more accurate camera poses than the first category.

Our method follows the PTAM [10] approach, but with the following changes. Firstly, we use a sparser set of more reliable feature points: we find Shi-Tomasi corners [23] in an input image (at the input scale only), order them by eigenvalue, then greedily select corners whilst ensuring a minimum distance from already selected corners. An example of the extracted features is shown in Fig. 2. Features are matched between frames using normalized cross-correlation. The camera pose of the second key frame and 3D feature locations are estimated in closed form by decomposing the essential matrix [11]. Camera pose is obtained by matching 2D features to triangulated 3D features, then solving the PnP problem using the non-linear method in [8]. However, we only use 3D features whose angle between the rays to the current view and the nearest key frame that feature was seen in is below a user defined threshold; this ensures that we do not match features that are unlikely to be visible.

An example on a toy data scene is shown in Fig. 3(b). The estimated 3D point cloud is sparse and does not provide sufficient detail. However, a denser reconstruction can be achieved by using the epipolar geometry to constrain feature matching to a 1D search along epipolar lines. This is discussed next.

2.2. Multi-view stereo

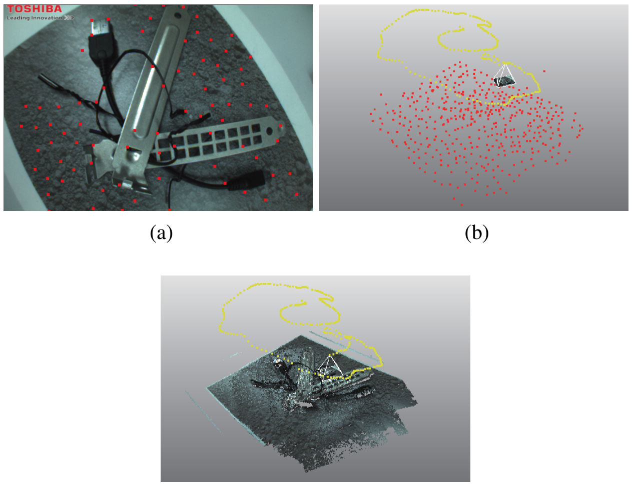
Multi-view stereo algorithms take as input calibrated images, and produce a point cloud or mesh as output. They can also be categorized into batch methods, which take in a number of frames and produce a single reconstruction [1, 7, 22], and online methods, which process a frame at a time, and output depth data as and when it is available [28, 30]. All these methods have a matching term, which is aggregated over several frames. Batch methods then tend to have a regularization step, which aggregates data across the entire image domain, which is why they compute a single reconstruction. By contrast, online methods tend to operate locally, so can reason about each



(a) PTAM feature points

(b) Our feature points

Figure 2: **Feature points for camera pose estimation.** *Input image with feature points computed by (a) PTAM (colour indicates scale features detected at), and (b) our feature detection method.*



(a)

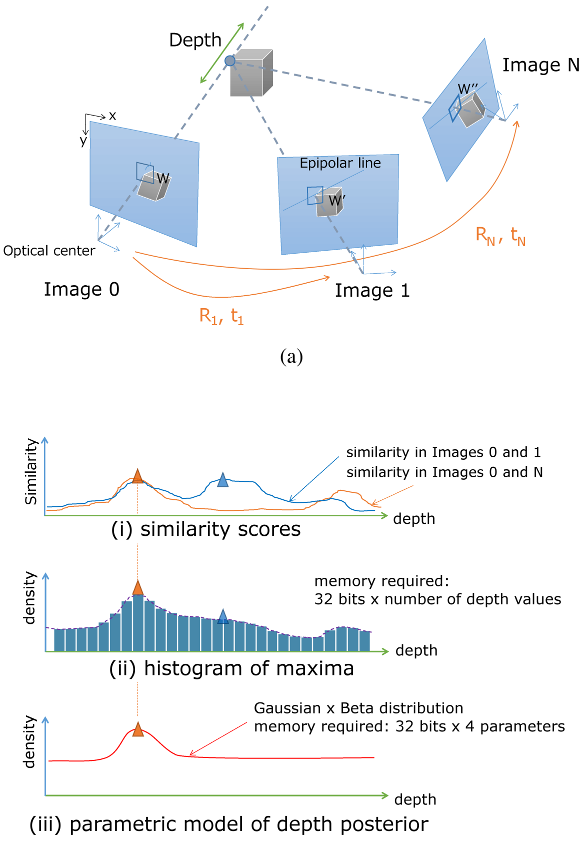
(b)

(c)

Figure 3: **3D reconstruction example on a toy data set.** *(a) Input image with feature points (red), (b) 3D view of feature points (red) and estimated camera trajectory (yellow), (c) 3D view of dense reconstruction.*

depth value independently, and provide a low-latency depth estimate—areas with high depth certainty can be output immediately. In terms of speed, batch methods are often slow due to the regularization step, which can involve a costly optimization, though several methods have been shown to run at interactive speeds. Online methods generally work faster, though not always at frame rate [30].

Due to the nature of our application, where accuracy is crucial and completeness is not, regularization is not so important because we do not wish to estimate the depth of less certain areas and risk getting it wrong. Also, a low-latency system is preferable, in order to provide the user with feedback as quickly as possible. For this reason we use the online approach in [28], which is outlined in Fig. 4. It initializes a set of image patches (we call them “seeds”) in a particular image, then matches these along their respective epipolar lines in subsequent images, using normalized cross-correlation. Matches are then used to update a probabilistic model for the depth of each patch, parameterized as a Gaussian distribution on inverse depth times a Beta distribution on the likelihood of being an inlier. This probabilistic approach both provides robustness to errors, and allows us to output a seed as a 3D point when it reaches a threshold depth accuracy and inlier probability. Because of the highly parallel nature of this algorithm, we implement it on a GPU, and match half a million seeds per frame.


Depth
Image N
W″
x
y
W
Epipolar line
W′
Optical center
Image 0
Image 1
R_N, t_N
R_1, t_1
Similarity
similarity in Images 0 and 1
similarity in Images 0 and N
depth
(i) similarity scores
density
memory required:
32 bits x number of depth values
depth
(ii) histogram of maxima
density
Gaussian x Beta distribution
memory required: 32 bits x 4 parameters
depth
(iii) parametric model of depth posterior

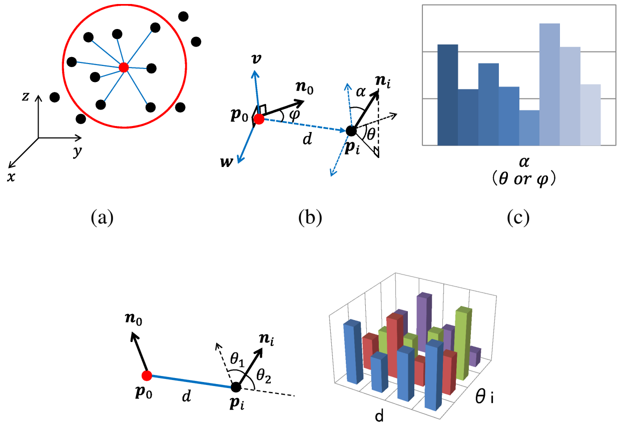
(a)

(b)

Figure 4: **Motion-based monocular stereo.** *(a) Reference view and two views after camera motion. Given camera poses, image patches are matched along their epipolar lines. Matching costs over several frames are aggregated, shown for a single pixel in (b), where the posterior distribution of depth for each pixel is modelled using a compact parametric model.*

2.3. Image-based relocalization

As described in 2.1, 3D camera motion is estimated by *tracking* features, therefore if the image motion is large



(a) (b) (c)

(d) (e)

Figure 5: **3D Feature Descriptors.** *Local descriptors are computed in a 3D neighbourhood around a point (a). The Fast Point Feature Histogram (FPFH) feature descriptor computes histograms over angles between pairs of normals (b, c) [20]. The proposed Co-occurrence Histogram of Angle and Distance (CHAD) feature descriptor (d, e) computes 2D histograms of quantities computed from normals and point distances.*

then tracking, and consequently camera pose estimation, can fail. This is a particular problem in our application as the radiation-proof camera has a narrow field of view, and the wind can cause significant camera, and therefore image, motion. In order to maintain a consistent reconstruction between tracking failures, fast and robust relocalization is required. We use image-based relocalization to recover camera pose, similar to the methods proposed in [12, 29], as this offers the lowest latency system possible, allowing relocalization from just one frame. Once the estimated camera motion exceeds a certain threshold in 3D position or angle, a new key frame is defined from which features are extracted [23] and added to a landmark database. For each feature in the key frame we compute an ORB descriptor [19] over four multi-resolution pyramid levels. When tracking fails, we find the closest database match for each feature in the new image based on the Hamming distance, which is efficiently computed using SSE instructions. Given the matches from 3D landmarks to 2D image points, we solve the resulting PnP problem to compute camera pose by non-linear optimization, initialized using the direct linear transform solution and non-linear optimization [26].

3. Point cloud registration

A post-processing stage of our reconstruction pipeline is to combine multiple point clouds, estimated from different video sequences, in a single, consistent coordinate frame. Point clouds are first crudely aligned, then the iterated closest point algorithm [2] is used to refine their relative pose. Approximate alignment is achieved by matching descriptors of 3D feature points, then using a hypothesize-and-test framework, here PROSAC [5], to compute a pose (using [27]) and reject mismatches. The feature points are extracted from each point cloud using a 3D extension of the standard 2D Harris detector, and descriptors are computed for each feature point.

3D point cloud descriptors can be categorized according to how they achieve rotational invariance. Some methods compute a robust local coordinate frame at each feature point, *e.g.* using PCA, then compute a rotationally variant descriptor in that coordinate frame. The Spin Image [9] is an example of this type. However, this approach relies on the repeatable computation of the local coordinate frame, which can be sensitive to noise. Other methods construct a descriptor that itself has rotational invariance. The Fast Point Feature Histogram (FPFH) [20] and our proposed method belong in this category. Fig. 5(b, c) give an overview of the FPFH descriptor. The descriptor is defined by a feature point $\mathbf{p}_0$ and points $\mathbf{p}_i$ in a local neighbourhood, see Fig. 5(a), as well as their normal vectors, $\mathbf{n}_0, \mathbf{n}_i$, respectively. The descriptor consists of 1D histograms of angles α, θ, and ϕ, describing the relative orientation of $\mathbf{n}_0$ and $\mathbf{n}_i$:

$$\begin{aligned} \alpha &= \mathbf{n}_i^T \mathbf{v}, \\ \theta &= \mathrm{atan2}(\mathbf{n}_i^T \mathbf{w}, \mathbf{n}_i^T \mathbf{n}_0), \\ \phi &= \frac{1}{d}(\mathbf{p}_i - \mathbf{p}_0)^T \mathbf{n}_0\,, \end{aligned} \tag{1}$$

where $d = \|\mathbf{p}_i - \mathbf{p}_0\|$, $\mathbf{v} = \mathbf{n}_0 \times (\mathbf{p}_i - \mathbf{p}_0)/d$, $\mathbf{w} = \mathbf{n}_0 \times \mathbf{v}$. Here we propose the Co-occurrence Histogram of Angle and Distance (CHAD) descriptor to capture local 3D shape. CHAD is a hybrid of the best parts of the Spin Image and FPFH descriptor methods. The 2D co-occurrence histogram of the Spin Image has higher descriptive power than the 1D histogram of FPFH, whilst the relative relation between two points of the FPFH preserves rotation invariance without requiring the computation of a local coordinate frame. Fig. 5(d, e) show an overview of the CHAD descriptor. We use the distance d between $\mathbf{p}_i$ and $\mathbf{p}_0$ as well as two angles: θ_1, between $\mathbf{n}_0$ and $\mathbf{n}_i$, and θ_2, the angle between $\mathbf{n}_i$ and the translation vector $(\mathbf{p}_i - \mathbf{p}_0)$:

$$\begin{aligned} \theta_1 &= \arccos\left(|\mathbf{n}_0^T \mathbf{n}_i|\right), \\ \theta_2 &= \arccos\left(\frac{1}{d}|(\mathbf{p}_i - \mathbf{p}_0)^T \mathbf{n}_i|\right), \\ d &= \|\mathbf{p}_i - \mathbf{p}_0\|. \end{aligned} \tag{2}$$

We create separate 2D co-occurrence histograms of d and θ_1 and of d and θ_2. The dimension of the CHAD descriptor is $D \times (T_1 + T_2)$, where D, T_1, and T_2 are the number of quantization bins of d, θ_1, and θ_2, respectively. Descriptor matches are found via exhaustive search using the Euclidean distance.

4. Component evaluation

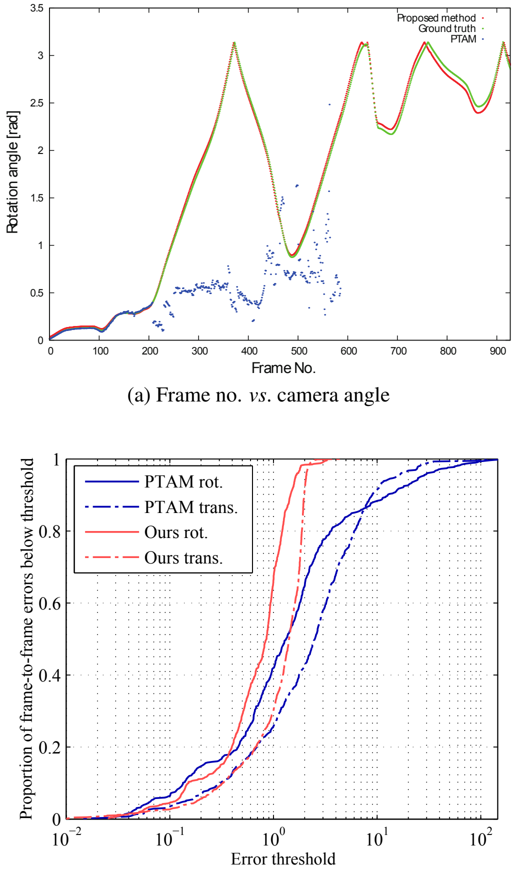
The system presented consists of several standard components, to which we have made some changes to improve performance. Whilst the main purpose of this system is the realtime reconstruction of the Fukushima Daiichi power station, for reasons of confidentiality we do not have access to the input data captured there, nor is there any ground truth available. Therefore, in order to provide a comparative evaluation of our system, we have evaluated the camera pose estimation, multi-view stereo and point cloud registration system components on standard datasets. The results are described below.

4.1. Camera pose estimation

We ran our proposed method on the synthetic Tsukuba stereo sequence [14, 15], which provides ground truth camera pose and scene depth. We use the publicly available implementation of PTAM [10] as a baseline, as our method is based on this approach. Results for both methods are shown in Fig. 6, where (a) shows that our proposed method tracks the ground truth camera angle well, with very little drift, whilst PTAM becomes error ridden after about 200 frames, and loses track completely before 600, and (b) shows that our method has a much lower proportion of large errors in both translation and rotation estimates between consecutive frames. This demonstrates the improved robustness provided by our feature selection and feature matching, described in section 2.1.

4.2. Multi-view stereo

We also evaluated our implementation of the real-time multi-view stereo method [28] on the Tsukuba dataset [14, 15], using the ground truth camera poses. Seeds are created in each frame, and a proportion of these are output as 3D points; the ratio of the two, completion rate, is plotted against frame number and also alongside camera rotation rate, in Fig. 7(c), showing that frames succeeded by large rotations have a lower completion rate. This is because large rotations tend to mean that seeds do not get matched in enough frames to meet the output criteria. Fig. 7(a) and (b) show results on the depth accuracy of output 3D points in the frames they were initialized in. Fig. 7(a) shows that there is a strong correlation between depth and depth error, but that depth error is in the region of 0.1 to 0.01 times the depth. Since depth is computed via triangulation, errors tend to be constant in inverse depth space. Fig. 7(b) shows that precision is high in this space, providing an accuracy of $2 \times 10^{-4}\text{cm}^{-1}$ or less for 70% of the points.



(a) Frame no. *vs*. camera angle

(b) Frame-to-frame errors

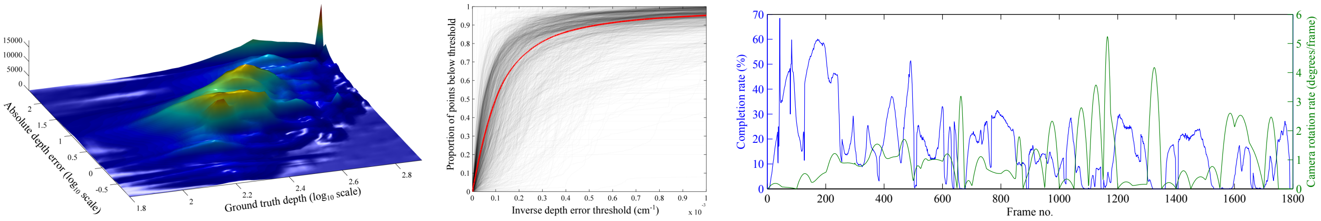
Figure 6: **Camera pose estimation accuracy.** *(a) A plot of camera rotation angle (relative to the first frame) across frames. (b) Error-proportion curves for frame-to-frame errors in rotation and translation.*

4.3. 3D Registration

We compared the performance of our CHAD descriptor for point clouds against the Spin Image and FPFH descriptors on which it is based. We used raw range data from the public Stanford Scanning Repository [18]. The dataset contains 3D point clouds of various objects captured from different view points.

In our experiments, descriptors are computed at 3D Harris corners. The descriptor sizes of Spin Image, FPFH, and CHAD are 153, 33, and 156, respectively.

Fig. 8(a) shows the matching performance of 3D feature points. On all data sets the proposed CHAD descriptor obtains the best performance, with the Spin Image and FPFH descriptors performing similar to each other on average. The matching rate is low on the "bun315", "top3", and "chin" point clouds because of the lower number of overlapping feature points. A match is defined by a threshold on the Euclidean distance between descriptor pairs. In the experi-

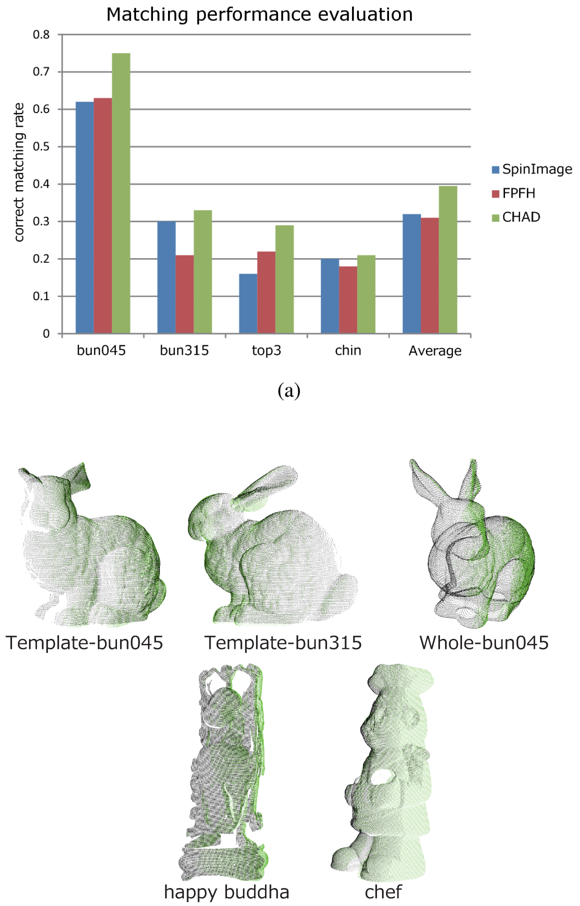


(a) 2D histogram of depth *vs*. error

(b) Per-frame precision-recall curves

(c) Completion rate (points output / seeds created), and camera rotation rate, *vs*. frame no.

Figure 7: **Multi-view stereo accuracy.** *(a) Depth errors across all 1800 frames of Tsukuba, histogrammed against ground truth depth. (b) Error-proportion curves in inverse depth space for all frames, with the average shown in red. (c) Completion rate (blue) and camera rotation rate (green) plotted against frame number.*



(a)

(b)

Figure 8: **Evaluation of 3D descriptors.** *(a) Correct matching percentages of different descriptors on range data from the Stanford Scanning Repository [18]. The proposed CHAD descriptor achieves a consistently higher number of correct matches. (b) shows 3D registration results on the same data set.*

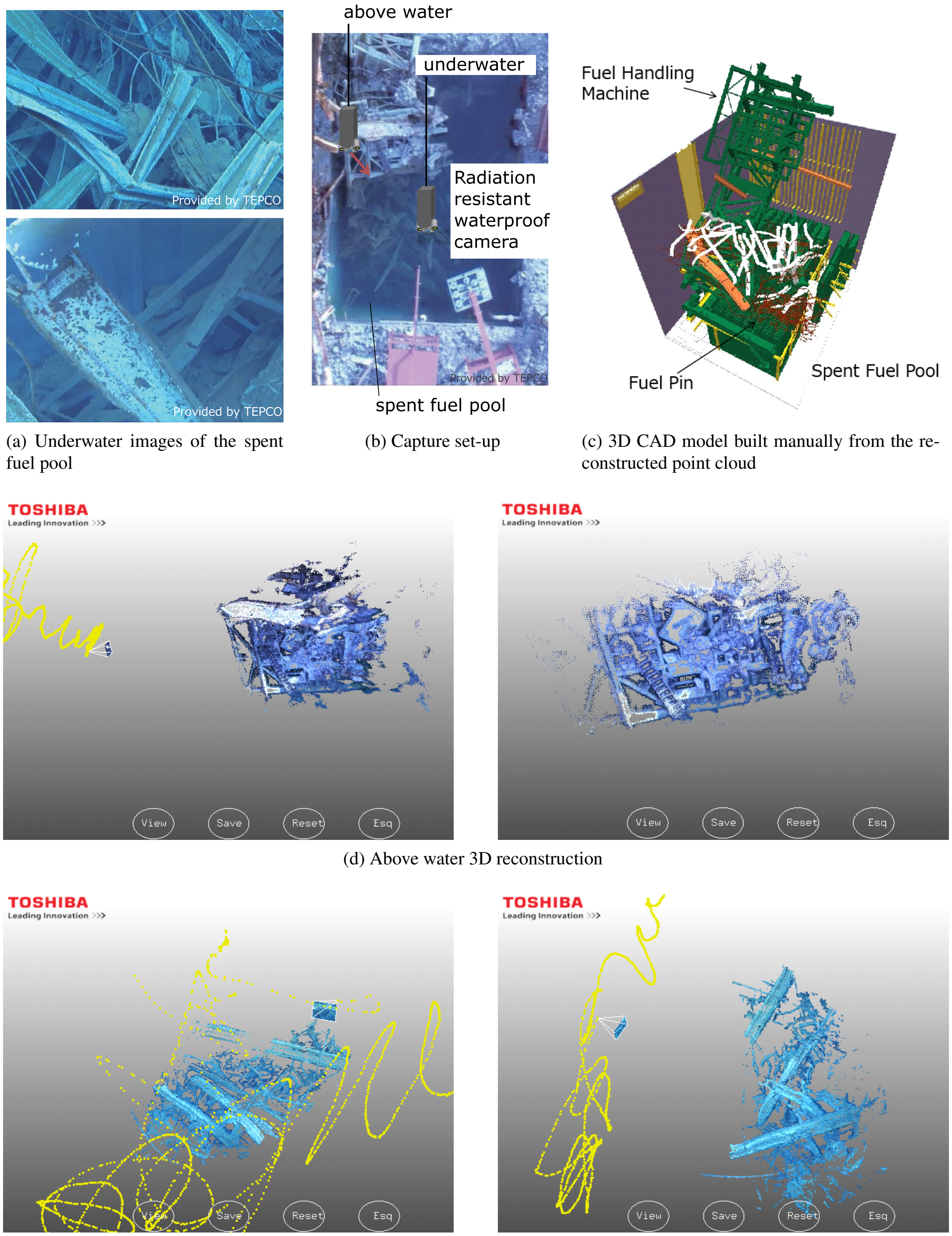
ments these thresholds are set for each descriptor such that the number of matched pairs is similar. Fig. 8(b) shows successful registration results using CHAD on range data, except for “Whole-bun045” which shows the “bun045” scan registered to the whole bunny 3D point cloud.

5. Fukushima reconstruction

In this section we present 3D reconstruction results captured at Fukushima Daiichi. Intrinsic parameters for the radiation-proof camera were computed in advance [3]. The underwater capture required separate calibration due to a different refraction index between the lens and water. Camera parameters were switched manually during the capture process. Fig. 9 shows reconstruction results captured at Fukushima Daiichi. The top row shows example frames from original underwater footage (a), the capture setup (b), and the final 3D CAD model (c). The bottom two rows show point clouds during the reconstruction process, above (d) and below (e) water, as well as the estimated camera trajectories. The trajectories indicate the swinging motion of the crane-mounted camera. The 3D CAD model (c) was created manually based on the dense 3D point clouds. It was used to estimate the centre of gravity of debris parts as well as their size and connectivity, thereby aiding the planning of their removal. The reconstruction system runs at 37ms per frame, implemented on a laptop with Intel®Core^{TM}i7 processor and Nvidia®GeForce®GTX^{TM}GPU.

6. Summary

This paper presented a system for assisting in the challenging clean-up project at the Fukushima Daiichi nuclear power station. We described four main system components, including point cloud registration with a new descriptor (CHAD), explained the design of each component, and provided quantitative evaluations of their performance. Finally we have provided qualitative results of the 3D reconstruction of the Fukushima Daiichi power station.



(a) Underwater images of the spent fuel pool

(b) Capture set-up

(c) 3D CAD model built manually from the reconstructed point cloud

(d) Above water 3D reconstruction

(e) Underwater 3D reconstruction

Figure 9: **Reconstruction of the spent fuel pool at Fukushima Daiichi.** *The system is used to reconstruct 3D structure from underwater scenes of a spent fuel pool (a) using a radiation and water proof camera (b). The final result is a CAD model (c), which is used to plan decommissioning work. It is created by manually fitting surfaces to reconstructed point clouds (d and e). Reconstructed points are shown in blue, the camera trajectories in yellow.*

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, pages 72–79, September/October 2009. 2

[2] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *TPAMI*, 14(2):239–256, 1992. 4

[3] J. Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/. 6

[4] I. Brilakis, H. Fathi, and A. Rashidi. Progressive 3D reconstruction of infrastructure with videogrammetry. *Automation in Construction*, 20(7):884–895, 2011. 1

[5] O. Chum and J. Matas. Matching with PROSAC progressive sample consensus. In *CVPR*, pages 220–226, 2005. 4

[6] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *TPAMI*, 26(6):1052–1067, 2007. 2

[7] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *TPAMI*, 32(8):1362–1376, 2010. 2

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 2

[9] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *TPAMI*, 21(5):433 –449, May 1999. 4

[10] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *ISMAR*, November 2007. 2, 5

[11] Z. Kukelova, M. Bujnak, and T. Pajdla. Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. In *BMVC*, pages 1–10, 2008. 2

[12] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele. Real-time image-based 6-DOF localization in large-scale environments. In *CVPR*, pages 1043–1050, 2012. 4

[13] J. Martinez-Carranza, A. Calway, and W. Mayol-Cuevas. Enhancing 6D visual relocalisation with depth cameras. In *IROS*, 2013. 1

[14] M. P. Martorell, A. Maki, S. Martull, Y. Ohkawa, and K. Fukui. Towards a simulation driven stereo vision system. In *ICPR*, pages 1038–1042, 2012. 5

[15] S. Martull, M. P. Martorell, and K. Fukui. Realistic CG stereo image dataset with ground truth disparity maps. In *Proc. ICPR Workshop TrakMark2012*, pages 40–42, 2012. 5

[16] E. Molinos, A. Llamazares, N. Hernndez, R. Arroyo, A. Cela, J. J. Yebes, M. Ocaa, and L. M. Bergasa. Perception and navigation in unknown environments: The DARPA robotics challenge. *Advances in Intelligent Systems and Computing*, 253, 2014. 1

[17] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *ICCV*, 2011. 2

[18] The Stanford 3D Scanning Repository. http://www-graphics.stanford.edu/data/3Dscanrep/. 5, 6

[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, pages 2564–2571, November 2011. 4

[20] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *ICRA*, May 2009. 4

[21] K. Sakurada, T. Okatani, and K. Deguchi. Detecting changes in 3D structure of a scene from multi-view images captured by a vehicle-mounted camera. In *CVPR*, 2013. 1

[22] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, volume 1, pages 519–526, 2006. 2

[23] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593–600, 1994. 2, 4

[24] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *Int. J. Comput. Vision*, 80(2):189–210, 2008. 2

[25] S. Stent, R. Gherardi, B. Stenger, K. Soga, and R. Cipolla. An image-based system for change detection on tunnel linings. In *Machine Vision and Applications*, pages 359–362, 2013. 1

[26] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2011. 4

[27] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *TPAMI*, 13(4):376–380, 1991. 4

[28] G. Vogiatzis and C. Hernandez. Video-based, real-time multi-view stereo. In *Image and Vision Computing*, volume 29, pages 434–441, 2011. 2, 3, 5

[29] B. Williams, G. Klein, and I. Reid. Real-time SLAM relocalisation. In *ICCV*, 2007. 4

[30] O. J. Woodford and G. Vogiatzis. A generative model for online depth fusion. In *ECCV*, 2012. 2, 3

Product names mentioned in this paper are trademarks of their respective companies.