Vol.2011-CVIM-176 No.12
2011/3/18

Multi-view 3D Reconstruction Techniques in Computer Vision

Yasutaka Furukawa[†1]

This paper gives introduction to 3D reconstruction techniques from multiple photographs in Computer Vision, a field known as 3D photography. The field of 3D photography has made dramatic progress in the last five years, and state-of-the-art techniques can now, for example, download millions of images of a city from Internet, find clusters of images that see the same parts of a city, estimate camera parameters, and recover their dense geometry. This paper gives high-level introduction to multi-view 3D reconstruction approaches and its three steps: 1) image acquisition; 2) camera parameter estimation; and 3) dense geometry reconstruction. The paper also introduces recent research projects and their reconstruction examples to illustrate the capabilities of state-of-the-art techniques.

1. Introduction

Automated acquisition of static 3D geometric information from images, a field known as 3D photography, has been an important problem in Computer Vision for many years. The quality of reconstruction has made dramatic progress in the last decade thanks to the developments of digital photography, yielding high-quality consumer grade cameras with reasonable prices, and the sophistication of 3D reconstruction algorithms. 3D photography not only has been a hot topic in the research community[18] but also given great influences to industry. For example, camera parameter estimation and 3D scene reconstruction is an indispensable process for augmented reality applications in an entertainment industry.[22] Photosynth[16] is a free online web-based service by Microsoft, where users upload photos, then the system automatically estimates camera poses, reconstructs scene geometry, and provides a photo browser that makes use of the 3D geometry information. Arc3D[21] is a similar free web-based service, where the goal of the system is not a photo browser, but dense 3D reconstruction and visualization of a scene, which produces impressive results. There exists commercial software, for example, *Photofly*[2] by Autodesk, which reconstructs dense 3D geometry and provides nice visualization from a set of user photos.

State-of-the-art research techniques also show impressive results (See Figure 1). Agarwal et al.[1] presented a large-scale system that downloads millions of images from Internet by using a search-keyword (e.g., *Rome*), finds clusters of images that see the same scenes (e.g., *Colosseum* or *Trevi Fountain*), estimates camera parameters, and recovers sparse scene geometry as a point cloud. Furukawa et al.[5] presented a large-scale dense reconstruction system that takes a set of calibrated photographs from Agarwal's system, which is as large as tens of thousands for a large scene, reconstructs dozens of million 3D points, and provides nice visualization of densely reconstructed scenes.

This paper provides an introduction to 3D reconstruction techniques that recovers static geometry information from multiple photographs. Multi-view reconstruction system usually consists of three steps: 1) image acquisition; 2) camera parameter estimation; and 3) dense geometry reconstruction. In this paper, we describe each of the three steps at high-level for introductory purposes, as the focus of the paper is not to describe details of state-of-the-art algorithms, but to provide basic understanding of an entire system and introduce latest research examples. The remaining portion of the paper is organized as follows. Section 2

†1 Google Inc.

© 2011 Information Processing Society of Japan

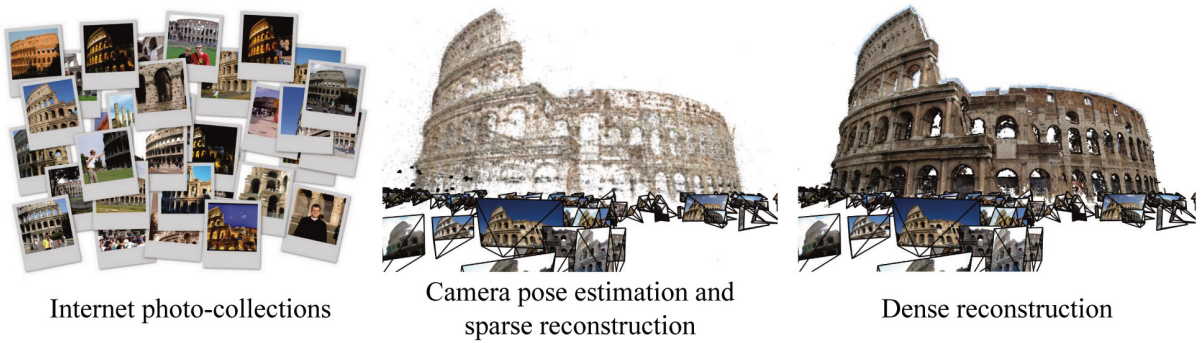

Fig. 1 State-of-the-art computer vision techniques can download millions of images from Internet, estimate camera poses, and recover dense 3D geometry information.

describes several methods and setups to acquire input images. An image acquisition method/setup restricts feasible camera parameter estimation methods, which are covered in Section 3. Section 4 describes multi-view stereo algorithms for dense reconstruction, and we conclude the paper in Section 5.

2. Image Acquisition

Image acquisition is the first important step for any image-based 3D reconstruction system. This section describes the following three typical image acquisition setups (See Figure 2).

The first setup is targeted for scanning a small-scale object in a lab environment, where a camera is fixed on a tripod and an object is placed on a turn table.[12] Photographs are taken while rotating an object on a turn table. In this setup, if a turn table is repetitive with high accuracy, an object of interests and a calibration chart can be photographed independently at separate times but under the same object motion. This makes it possible to calibrate cameras from images of a calibration chart (See Section 3.1 for more details). A similar, but more expensive, solution is to fix an object, but attach a camera to a robot arm, whose control information automatically yields camera parameters for each photo without any post-processing. ⋆1 Although robot arms are usually very expensive and not easy to come-by, one of its advantages over the turn-table is that an object is fixed and lighting condition stays the same with respect to an object, which makes texture analysis and 3D reconstruction easier.

The second setup is more flexible and can be used outside a lab environment, where you carry a camera, move around and take photos of an object or a scene. This is suitable for capturing outdoor scenes (e.g., buildings), however it is usually difficult to use calibration chart for such large scenes, and camera parameters need to be directly estimated from the input images with a Computer Vision algorithm (See Section 3.1 for more details).

The last setup, or an option, is to download images from In-

⋆1 This setup has been used to acquire datasets for a multi-view stereo evaluation project.[18]

© 2011 Information Processing Society of Japan

Fig. 2 Typical image acquisition setups. Left: Camera locations are fixed and an object is placed on a turn-table. Right: A scene or an object is fixed outdoors and you move around with a camera and take photos.

ternet by using, for example, community photo sharing websites such as Flickr (`http://www.flickr.com`). You need not even use a camera for this option, on the other hand, has no control over captured objects or scenes. Therefore, camera parameter estimation must be done through a Computer Vision system. Note that online community photos tend to have varying illuminations, viewpoints, image qualities and etc., which pose challenges to 3D reconstruction algorithms.[1),10)]

3. Camera Parameter Estimation

Given a set of images, the next important step is to estimate camera parameters for each image, which consist of extrinsic, intrinsic and distortion parameters. Extrinsic parameters contain rotational and translational pose information of the camera and change when you physically move a camera body. Intrinsic parameters contain information such as a pixel censor size, a principal point, and magnification factors. Intrinsic parameters are determined at the manufacturing stage, except for the magnification factors that change when a focus changes. Distortion parameters capture higher-order (non-linear) effects that come from lens. Detailed explanation of the camera parameters is outside the scope of the paper and interested readers are referred to a computer vision text book by Hartley and Zisserman.[11)]

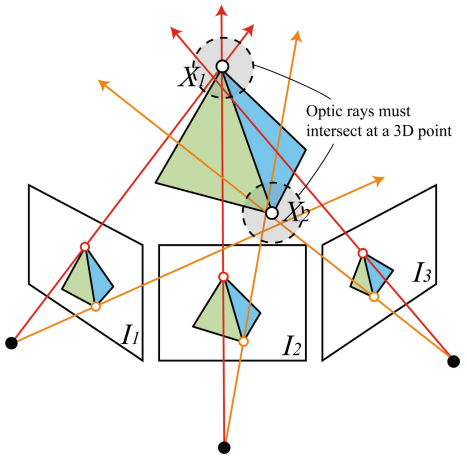

Fig. 3 Camera parameters can be estimated by feature correspondences across multiple images. Suppose we identify the projected location of a vertex X_1 (resp. X_2) in every image, and establish their correspondence. Each set of matched feature points poses a constraint on the camera parameters in that optic rays passing through the projected image locations (e.g., three red rays) must intersect at a single 3D point.

While there are various camera calibration methods, a common essential task is to extract image features (e.g., corners and blobs) and establish their correspondences across images. Figure 3 illustrates a toy example where a tetrahedron is visible in three images $\{I_j\}$. Suppose we identify and extract image locations of the vertices and match them across the images. Then, each vertex imposes a constraint on the camera parameters in that

© 2011 Information Processing Society of Japan

optic rays that pass through its extracted image locations must intersect at a single 3D point. For example, in Figure 3, the three red rays (resp. the three orange rays), corresponding to a vertex X_1 (resp. X_2), must intersect at a single 3D point. Therefore, the estimation process is to extract and match image features as many as possible and solve for camera parameters that satisfy the above intersection constraint for all the matched feature points.

In the following, we describe two different camera calibration methods, calibration chart-based system and Structure from Motion, whose main difference is the way image features are detected and matched. We again refer interested readers to a text book[11] for algorithmic details.

3.1 Calibration chart based system

For image acquisition in a lab environment, an access to a photographed scene is allowed. In such a scenario, a calibration chart can be used to help calibrate cameras: We know the geometry and texture of a calibration chart, and feature extraction and correspondence computation becomes easy.

A typical experimental setup is to use a turn table that can repeat its motion with high accuracy. We take photos of an object and a calibration chart on a turn table separately but under the same motion. Then, we calibrate cameras with photos of a calibration chart by exploiting its known geometry and texture information. Jean-Yves Bouguet distributes matlab calibration software with friendly GUI interface for a planar checker-board calibration chart.[3] ⋆1 A user needs to simply click three corners of the checker-board pattern in each photo, where the system automatically detects and establish correspondences of all the grid corners and estimate the camera parameters.

⋆1 C++ implementation of the software is also available from open source computer vision library OpenCV.[9]

3.2 Structure from Motion

When a user does not have an access to photographed scenes or it is difficult to place calibration chart, Structure from Motion (SfM) is a popular alternative. SfM is a process of recovering 3D structure of a scene and camera parameters[11] automatically from a set of images. SfM system can be used to process an image sequence (e.g., movie shots)[14),22] as well as photo collections.[2),16] When an image sequence is the input, feature detection and tracking method is used to establish feature correspondence.[14),22] For a separated photo collection, feature points are first detected by feature detectors (e.g., corner or blob detectors), processed by feature descriptors such as SIFT[15] to make them robust against viewpoint and illumination changes, and matched across images.

SfM system often works as a black box, where users need to just provide an image sequence or photos as input, and is very easy to use especially for non-experts. On the other hand, input data must be acquired with certain considerations for robust and successful reconstructions. For an image sequence, motion blur, and hence fast motion, should be avoided for image quality. For an image set, there must be enough image overlap with sufficient but not too large viewpoint changes. Furthermore, in comparison to calibration chart, scene geometry and appearance are unknown, which poses additional challenges to SfM algorithms.

Nonetheless, successful SfM system for an image sequence has been developed and become an indispensable process in an entertainment industry (e.g., movie production). In a research community, SfM on a photo collection has been a very popular and successful field[1),4),20] thanks to emerging internet photo collections such as Flickr (`http://www.flickr.com`).

Bundler[19] is a state-of-the-art SfM software by Noah Snavely, which originates from Photo Tourism project at University of

© 2011 Information Processing Society of Japan

Fig. 4 An SfM reconstruction of Colosseum in Rome from internet community photo collections by Noah Snavely's SfM software *Bundler*.[19)]

Washington.[20)] Photo Tourism was the first successful SfM system on internet photo collections (See Figure 4), and became the basis for Photosynth from Microsoft Live Labs.[16)] The software is unfortunately not scalable and can process at most hundreds of photos. Large-scale SfM algorithms have been recently presented in the research community. Agarwal et al. presented a system[1)] that downloads millions of images from an online community photo-sharing website Flickr (`http://www.flickr.com`) by using a search keyword such as *Rome* or *Venice*, finds clusters of images that see the same scenes such as *Colosseum* or *Trevi Fountain* in Rome and *San Marco Square* or *Grand Canal* in Venice, estimates camera pose, and reconstructs 3D points corresponding to matched image feature points (See Figure 5). The system produced, for example, a cluster of 2,097 images and 819,242 points of Colosseum in Rome, a cluster of 13,699 images and 4,515,157 points for San Marco Square in Venice, and 4,585 images and 2,662,981 points for a city of Dubrovnik in Croatia. The system runs less than a day for each city with a PC cluster with 500 compute cores. Frahm et al. also presented a large-scale SfM system that can process a couple million Internet photos within a day on a single PC, while achieving significant speed-ups with GPU-based implementation.[4)]

4. Dense Geometry Reconstruction

In addition to estimating camera parameters, camera calibration process recovers a set of 3D points. However, these points are usually very sparse corresponding to only distinctive features in the images, and noisy. Reconstructing dense and clean 3D points or mesh models from calibrated photographs is addressed by multi-view stereo (MVS) algorithms.

MVS algorithms have made dramatic progress in the last few years, and state-of-the-art algorithms can now rival laser range scanners in accuracy. ⋆1 In the following, we first explain a basic MVS principal, namely photometric consistency function, then introduce notable research work in the field.

4.1 Photometric consistency function

MVS algorithms use a *photometric consistency function* as a fundamental tool for 3D reconstruction. Given a 3D point X_i, a photometric consistency function can be evaluated as follows. We first project X_i to each input image, collect a pixel color at its image projection, then check the consistency of sampled pixel colors by taking their standard-deviation, for example. If a point lies on the surface of an object or a scene, collected pixel colors come from the same physical 3D point and should be consistent (e.g., X_2 in Figure 6). If not, sampled pixel colors should come from different parts of a scene and be inconsistent (e.g., X_1 in Figure 6).

Suppose a photometric consistency function is evaluated at every 3D point in a 3D volume. In a toy example illustrated at the right of Figure 6, a photometric consistency function is expected

⋆1 Seitz et al. gives a good survey on MVS algorithms.[18)]

© 2011 Information Processing Society of Japan

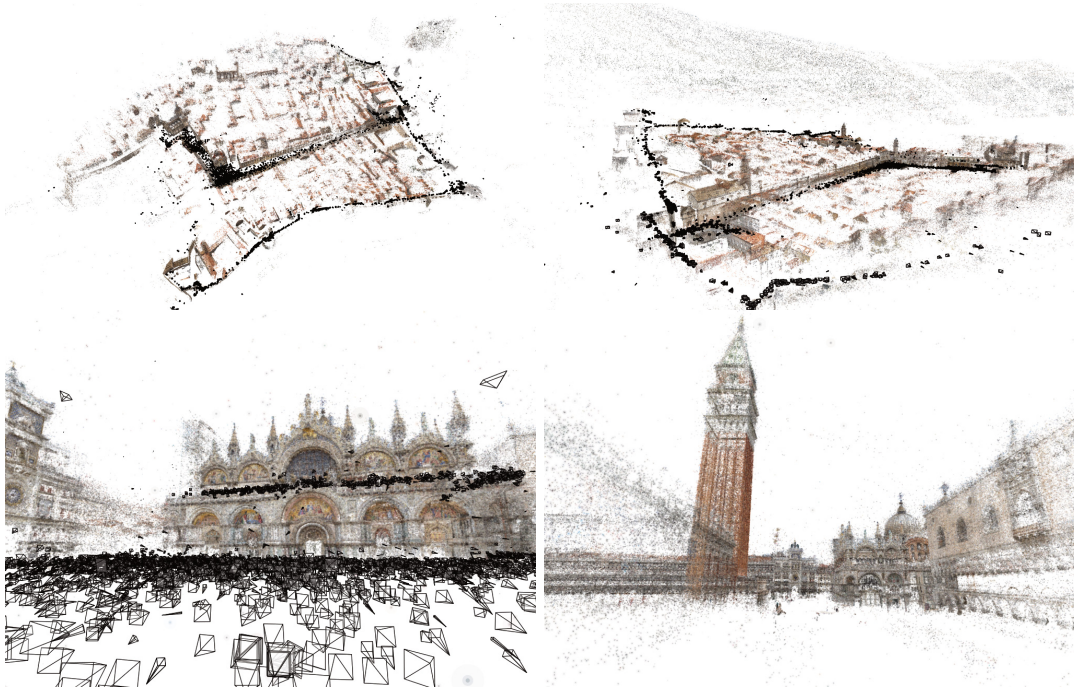

Fig. 5 Reconstruction results of large-scale SfM system.[1] Top: A city of Dubrovnik in Croatia with 4,585 images and 2,662,981 3D points. Bottom: San Marco Square in Venice with 13,699 images and 4,515,157 3D points. In the figure, each estimated camera pose is represented by a frustum (line-drawing), while reconstructed points are rendered as colored 3D points in the background.

to be high (black in the figure) near the surface of an object. MVS reconstruction is a problem to identify a 2D surface in the volume where the consistency score is high. Note that the above is a description of a very simple photometric consistency function, and numerous variants have been proposed in the past. Refer to a survey paper by Seitz et al.[18] for more details.

4.2 State-of-the-art Algorithms

Esteban et al. presented the first successful MVS approach that produces high-quality mesh models six years ago (See Figure 7).[12] As in Esteban's work, most MVS algorithms aim at reconstructing scenes as polygonal mesh models, while many algorithms also use different surface representations in intermediate steps. Multiple-depthmap is a popular surface representation, where a depthmap is reconstructed for each image,[8),23] which are then merged into a single 3D model (See Figure 8). An oriented point cloud is another representation,[7] which in fact suffices for

© 2011 Information Processing Society of Japan

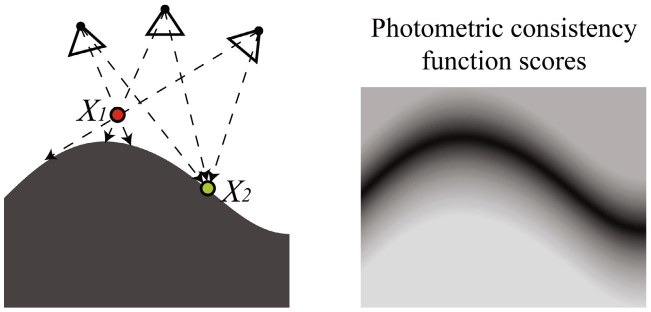

Fig. 6 Photometric consistency function evaluates if a 3D point is likely to be on the surface of an object or a scene. Given a 3D point, we project the point into each image and collect a pixel color at its projected image location. Collected pixel colors in multiple images should be consistent if the point is on the surface. MVS reconstruction problem is to identify a 2D surface where the photometric consistency score is high, illustrated as black region at the right of the figure.

Fig. 7 High quality MVS mesh models reconstructed by Esteban et al.[12] in 2004. A couple dozen high resolution photographs are acquired by a turn table-based setup (left). All the figures are courtesy of Carlos H. Esteban.

visualization purposes[5] via point-based rendering techniques,[17] but can also be converted into a mesh model (See Figure 9). ⋆1

⋆1 Patch-based Multi-View Stereo (PMVS) by Furukawa et al.[6] is a popular MVS software that produces oriented point clouds, and Poisson Surface reconstruction software by Kazhdan et al.[13] converts oriented points into a polygonal mesh model. Both software are open-source and publicly available online.

Fig. 8 Real time MVS system based on multiple depthmap surface representation by Gallup et al.[8] The figure is courtesy of David Gallup.

Fig. 9 Oriented point clouds reconstructed from patch-based multi-view stereo algorithm by Furukawa et al.[7] (top), which are then converted into mesh models (bottom).

Most multi-view stereo researches have focused on small-scale object or scene reconstruction involving in a couple hundred photographs at most. Furukawa et al. presented a large-scale MVS

© 2011 Information Processing Society of Japan

system[5] that was built on Agarwal's SfM system described in Section 3.2. The core challenge is in the view clustering, namely process to decompose an input image set into smaller image clusters, for which a standard MVS algorithm is used to reconstruct a scene independently. Figure 10 shows two reconstructions from the system. The top of the figure shows 14,051,331 3D points for an old city of Dubrovnik from 6,304 input images. The bottom shows reconstructed 27,707,825 3D points for San Marco Square in Venice from 13,709 images. The system runs in a few hours via parallel execution on distributed system. In both examples, the system succeeded in reconstructing wider areas, while achieving high fidelity at popular locations where images are abundant.

5. Conclusion

In this paper, we gave an end-to-end introduction to 3D photography techniques and system in Computer Vision. With recent advancements and maturity in the research field, more vision-based 3D reconstruction system have been deployed in industry, and more 3D reconstruction tools have become available for non-experts. Not so far distant in the future, there may come a day when an entire world can be reconstructed from everybody's photo collections.

References

1) Agarwal, S., Snavely, N., Simon, I., Seitz, S.M. and Szeliski, R.: Building Rome in a Day, *ICCV* (2009).
2) Autodesk: Project Photofly, `http://labs.autodesk.com/utilities/photo_scene_editor/`.
3) Bouguet, J.-Y.: Camera Calibration Toolbox for Matlab, `http://www.vision.caltech.edu/bouguetj/calib_doc`.
4) Frahm, J.-M., Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S. and Pollefeys, M.: Building Rome on a Cloudless Day, *ECCV* (2010).
5) Furukawa, Y., Curless, B., Seitz, S.M. and Szeliski, R.: Towards Internet-scale Multi-View Stereo, *CVPR* (2010).
6) Furukawa, Y. and Ponce, J.: PMVS, `http://www.cs.washington.edu/homes/furukawa/research/pmvs`.
7) Furukawa, Y. and Ponce, J.: Accurate, Dense, and Robust Multi-View Stereopsis, *PAMI*, Vol.32, No.8, pp.1362–1376 (2010).
8) Gallup, D., Frahm, J.M., Mordohai, P., Yang, Q. and Pollefeys, M.: Real-time Plane-sweeping Stereo with Multiple Sweeping Directions, *CVPR* (2007).
9) Garage, W.: OpenCV, `http://opencv.willowgarage.com/wiki`.
10) Goesele, M., Snavely, N., Curless, B., Hoppe, H. and Seitz, S. M.: Multi-View Stereo for Community Photo Collections, *ICCV* (2007).
11) Hartley, R.I. and Zisserman, A.: *Multiple View Geometry in Computer Vision*, Cambridge University Press (2004).
12) Hernández Esteban, C. and Schmitt, F.: Silhouette and stereo fusion for 3D object modeling, *CVIU*, Vol.96, No.3, pp.367–392 (2004).
13) Kazhdan, M., Bolitho, M. and Hoppe, H.: Poisson Surface Reconstruction, *Symp. Geom. Proc.* (2006).
14) Laboratoriumfr Informationstechnologie, U. o.H.: Voodoo Camera Tracker: A tool for the integration of virtual and real scenes, `http://www.digilab.uni-hannover.de/docs/manual.html`.
15) Lowe, D.: Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, Vol.20, pp.91–110 (2003).
16) Microsoft: Photosynth, `http://photosynth.net`.
17) Rusinkiewicz, S. and Levoy, M.: QSplat: A Multiresolution Point Rendering System for Large Meshes, *SIGGRAPH* (2000).
18) Seitz, S.M., Curless, B., Diebel, J., Scharstein, D. and Szeliski, R.: A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms, *CVPR*, Vol.1, pp.519–528 (2006).
19) Snavely, N.: Bundler: SfM for Unordered Image Collections, `http://phototour.cs.washington.edu/bundler`.
20) Snavely, N., Seitz, S.M. and Szeliski, R.: Photo tourism: exploring photo collections in 3D, *SIGGRAPH* (2006).
21) Vergauwen, M. and Gool, L.V.: Web-based 3D Reconstruction Service, *Mach. Vision Appl.*, Vol.17, No.6, pp.411–426 (2006).
22) VICON: Boujou, `http://www.vicon.com/boujou`.

© 2011 Information Processing Society of Japan

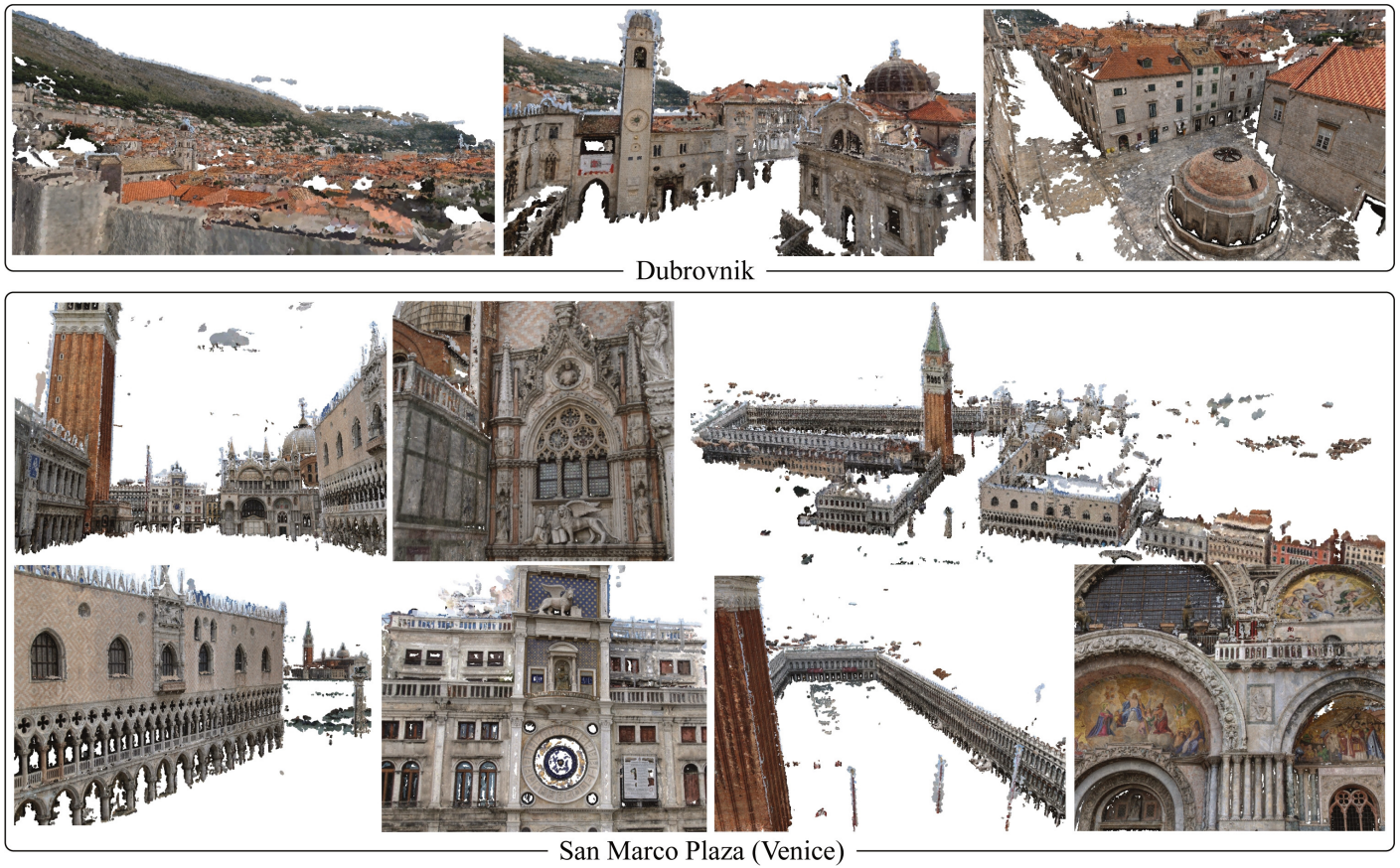

Fig. 10 Reconstruction results from scalable MVS system by Furukawa et al.[5] Top: 14,051,331 points are reconstructed from 6,304 images for an old city of Dubrovnik. Bottom: 27,707,825 3D points are reconstructed from 13,709 images for San Marco Square in Venice.

23) Zach, C., Pock, T. and Bischof, H.: A Globally Optimal Algorithm for Robust TV-L^1 Range Image Integration, *ICCV* (2007).

© 2011 Information Processing Society of Japan