Kim H, Evans A, Blat J, Hilton A. Multimodal Visual Data Registration for Web-Based Visualization in Media Production. IEEE transactions on circuits and systems for video technology. 2018;28(4):863–77. DOI: 10.1109/TCSVT.2016.2642825

Multimodal Visual Data Registration for Web-Based Visualization in Media Production

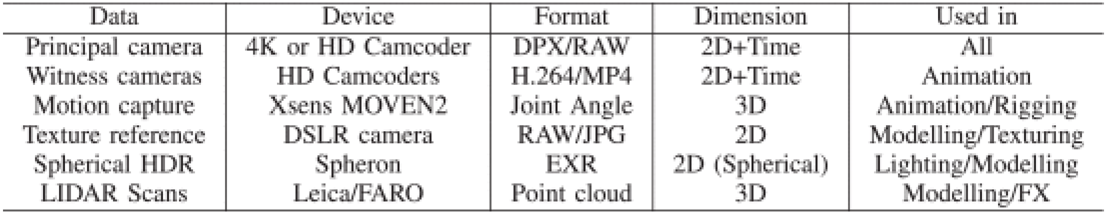
Abstract:
Recent developments of video and sensing technology have led to large volumes of digital media data. Current media production relies on videos from the principal camera together with a wide variety of heterogeneous source of supporting data [photos, light detection and ranging point clouds, witness video camera, high dynamic range imaging, and depth imagery]. Registration of visual data acquired from various 2D and 3D sensing modalities is challenging because current matching and registration methods are not appropriate due to differences in structure, format, and noise characteristics for multimodal data. A combined 2D/3D visualization of this registered data allows an integrated overview of the entire data set. For such a visualization, a Web-based context presents several advantages. In this paper, we propose a unified framework for registration and visualization of this type of visual media data. A new feature description and matching method is proposed, adaptively considering local geometry, semiglobal geometry, and color information in the scene for more robust registration. The resulting registered 2D/3D multimodal visual data are too large to be downloaded and viewed directly via the Web browser, while maintaining an acceptable user experience. Thus, we employ hierarchical techniques for compression and restructuring to enable efficient transmission and visualization over the Web, leading to interactive visualization as registered point clouds, 2D images, and videos in the browser, improving on the current state-of-the-art techniques for Web-based visualization of big media data. This is the first unified 3D Web-based visualization of multimodal visual media production data sets. The proposed pipeline is tested on big multimodal data set typical of film and broadcast production, which are made publicly available. The proposed feature description method shows two times higher precision of feature matching and more stable registration performance than existing 3D feature descriptors.

SECTION I.

Introduction

The development of visual sensor technology over recent decades has led to various 2D/3D media content acquisition devices available in our lives. In digital media production, broadcasting, game design, or virtual/augmented reality systems, the trend is to deal with big data captured not only from video or photography but also from a variety of digital sensors. The appearance of a scene can be captured using different digital video cameras, from 4K/6K and professional HD cameras, to those of mobile phones. Time-of-flight or Kinect-like RGBD sensors can capture video-rate depth information, while 3D laser scans create a dense and accurate geometrical point cloud of the scene. Spherical high dynamic range imaging scanners capture full 360° texture and illumination data, which is important for backplates and relighting. There may be other data sources such as video capture using drones or large collections of images captured with high-resolution DSLR cameras. There is an explosion in the volume, variety, and complexity of data that outstrip the capacity of current methods to manage, analyze, and visualize them. In digital production, it is typical for a single film to use >1 PB of storage for media assets with requirements increasing year-on-year. For example, 350 TB was allocated to the footage from various capture devices for the production of *John Carter of Mars* (2012), and *Avengers: Age of Ultron* (2015) is reported to have required >1 PB of storage. The types of data that are typically captured using visual sensors for film production, games, VR experience, and TV production are shown in Table I. While data storage is cheaper than ever, all of these data need to be sorted, indexed, and processed, which is a largely manual task.

TABLE I Examples of Data Types Generated in Film Production



Data	Device	Format	Dimension	Used in
Principal camera	4K or HD Camcoder	DPX/RAW	2D+Time	All
Witness cameras	HD Camcoders	H.264/MP4	2D+Time	Animation
Motion capture	Xsens MOVEN2	Joint Angle	3D	Animation/Rigging
Texture reference	DSLR camera	RAW/JPG	2D	Modelling/Texturing
Spherical HDR	Spheron	EXR	2D (Spherical)	Lighting/Modelling
LIDAR Scans	Leica/FARO	Point cloud	3D	Modelling/FX

We previously presented a multiple HD video camera system for studio production [1], which addressed the registration of multiple cameras to the world coordinate through calibration for 3D video production of actor performance. This has been extended to outdoor capture by combining multiple HD cameras and a spherical camera [2]. Dynamic objects captured by HD video cameras and static background scene scanned by a spherical camera were registered to the world coordinate system. In this paper, we extend the capture system further to allow automatic registration of the wide variety of visual data capture devices typically used in production.

A key issue is automatic registration of multimodal visual data into a common coordinate system to allow visualization and verification of the completeness of the data. This is essential to validate data collection at the point of capture. The task of handling 3D data is not merely a case of extending the dimensionality of existing 2D image processing. Data matching and registration is more difficult because 3D data can exist in different domains with different types of format, characteristics, density, and sources of error. In this paper, we introduce a unified 3D space (Fig. 1), where 2D and 3D data are registered for efficient data management and visualization. 2D data are registered via 3D reconstruction because direct registration of 2D to 3D structure [3], [4] is difficult to be applied for general multimodal data registration. We assume that multiple 2D data exist for the same scene so that 3D geometric information can be extracted.

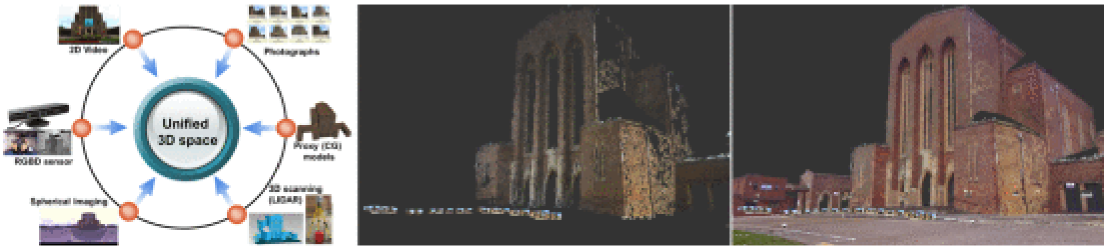


Fig. 1.
Multimodal data registration and visualization. Left: overview of multimodal visual data registration. Middle: multiple photographs and their 3D reconstruction. Right: registration to LIDAR coordinate system.

This unified space grounded in registration should be visualized integrating multiple 2D data (for example, video footage from several cameras) with raw 3D data (for example, laser-scan point clouds). A Web (or browser)-based application permits seamless mixing of 2D and 3D in a single context, allowing users to more quickly understand and navigate through the scene [5]. A Web application has further advantages: it is platform independent, accessible remotely, and easy to update and maintain. It requires no external software to be installed, is suited for access from all over the world, and supports collaborative workflows. In this sense, there is a strong drive for many modern visualization applications to be Web-based [6]. However, it requires great care in both its design and implementation, as a poorly designed hybrid 2D–3D visual experience can be incoherent in its use, and awkward to create. On the other hand, the raw multimodal data discussed in this paper is large (and thus difficult to transfer over the Web), and by its very nature has no consistent format or structure. Web-based 3D rendering is an emerging subject which has recently reached a new level of maturity, with recognition that the challenges faced are considerably different to those of offline rendering [7]. The most relevant issues are the time taken to download the data set to a remote client, and the challenge of visualizing such big data in a (relatively underpowered) Web browser. This paper directly addresses these challenges: the combination of modalities, the efficient use of bandwidth, and the processing at the client side (which has implications on usability).

The following are the main contributions of this paper:

1. a complete system from capture to visualization through data processing and transfer for efficient management of multisensory visual data from 3D and 2D modalities;

2. a robust multimodal visual data registration method using a multidomain (color, local geometry, and semiglobal geometry) feature descriptor and hybrid RANSAC-based matching method;

3. comprehensive evaluation of 3D feature detectors and descriptors for registration of 3D data of the built environment from multiple visual sensors;

4. a progressive, level-of-detail (LOD) Web-based visualization of multimodal visual data sets for efficient data transfer and interactive rendering;

5. a public multimodal database captured with a wide variety of devices in different environments to assist further research.

SECTION II.

Related Work

A. Multimodal Visual Data Registration

In general visual media processing, there has been some research for 2D/3D data matching and registration via structure-from-motion (SfM) and feature matching. 2D–3D registration between two modalities such as images to light detection and ranging (LIDAR) [4], [8], [9], images to range sensor [10], [11], or spherical images to LIDAR [12], [13] has also been investigated. To the best of our knowledge, 2D and 3D data registration and visualization for three or more visual modalities in general environments had not been investigated until our preliminary research. Initially, we tested existing 3D feature descriptors on multimodal registration [14] and applied them to different domains (local, keypoint, and color domains) in order to verify the influence of color and feature geometry [15]. The work in this paper goes far beyond our previous works. We propose a full 2D/3D multimodal data registration pipeline from capture to visualization using multidomain feature description and hybrid RANSAC-based registration based on the observations from our preliminary research. The proposed algorithms are tested on public multimodal data sets, and objective analysis of feature matching and registration performance is provided in this paper.

B. 3D Feature Detection and Descriptors

Feature (keypoint) detection identifies the location of distinct points in terms of variation in data. There have been many 3D keypoint detectors developed and evaluated for high distinctiveness and repeatability on 3D point clouds [16], [17]. However, the majority of the best performing detectors are not suitable for multimodal data registration because source models can have different color histograms, or errors in their geometry, according to the characteristics of the capture device. We prefer classic detectors which produce a relatively large number of evenly distributed keypoints such as Kanade-Tomasi detector [18] used in our previous research.

Feature descriptors define the characteristics of keypoints. Restrepo and Mundy [19] tested local 3D descriptors for registering 3D point clouds reconstructed by multiview stereo methods. Recently, Guo *et al*. [20], [21] performed a comprehensive evaluation of local feature descriptors on various data sets from different modalities, but the test was carried out not across modalities as in this paper but only within single modality in each data set. We performed similar evaluation to Restrepo and Mundy's work on multimodal data [14] and found that fast point feature histograms (FPFH) [22] and signature of histograms of orientations (SHOT) [23] descriptors are the most appropriate for multimodal data registration. In [24] and [25], cascade combination of shape and color descriptors showed good performance when the color information is available. However, color information is not always trusted in multimodal 3D data captured by different sensors because it is difficult to balance colors between modalities. Appearance information cannot be trusted for non-Lambertian surface or repetitive patterns. The descriptors are concatenated without any priority or weight in [24] and [25], which leads to poor performance when the matching is dominated by one descriptor as demonstrated in our preliminary research [15]. In this paper, we propose a novel matching and registration algorithm adaptively considering multiple descriptors using a hybrid RANSAC technique.

C. Web-Based Visualization of Multimodal Data

Jankowski and Decker [5], [26] demonstrated that a so-called "dual-mode" interface, integrating text and 3D contexts, outperforms a more classical approach where they are separated (even taking into account modality switches). The visualization of the unified space proposed in this paper requires such hybrid integration of 3D (more challenging than that of Jankowski), and a wealth of layered 2D data and metadata. An HTML5 Web context is suitable, in this regard, as it allows (and indeed encourages) the interplay of multimedia data. 3D Web pages are relatively uncommon (compared to 2D pages), and for several years were mostly represented by declarative technologies developed in the academic domain [27], [28]. However, 3D Web applications have been growing in popularity since the release of WebGL in 2011. WebGL is a Web-specific version of the OpenGL graphics API (more specifically of the restricted embedded systems API, OpenGL ES2.0), and allows access to dedicated graphics processing hardware directly from the browser (via Javascript). It is now fully supported in the latest versions of all major browsers. WebGL and associated HTML5 APIs (such as WebAudio)[1] are in many respects enabling technologies, as they break down the barriers for the development of browser-based multimedia applications. Nevertheless, they also opens up new research challenges for the best way to transmit and interact with hybrid data (be it 3D, 2D image/video, audio, or text).

3D data are typically large, and transferring it to a remote client for rendering is a persistent problem for all Web 3D applications. This is particularly relevant for our work in that the multimodal visual data are stored in files which reach many hundreds of megabytes in size—simply "waiting for them to download" does not provide an optimal or satisfactory user experience. While a naive approach might be to simply compress the data using any number of established and powerful algorithms, Limper *et al*. [29] show that straightforward data compression may not necessarily be the solution, as the decompression time in a browser-based context may outweigh any benefits gained in terms of compressed data, particularly as bandwidth speeds increase. For a more complete overview of these issues, and the current state of the art with respect to Web-based 3D, including techniques of remote rendering and progressive transmission, we refer the reader to a recent survey paper [7].

In our preliminary research in this field, we presented a similar progressive visualization of large point cloud data, where the data are preprocessed, in an offline step, into a hierarchical data structure [30]. Web-based rendering of very large point-clouds is tackled in [31], which uses a LOD approach to ensure the number of points rendered does not saturate the browser application. Only a single point cloud visualization in a Web environment was dealt with in [30], but we present algorithms and interface for the simultaneous visualization of multiple point clouds, intertwined with 2D image and video data in a single Web-based visualization platform in this paper. The results compare favorably for transmission times for the different but related problem of mesh visualization in [32].

SECTION III.

System Overview

Fig. 2 shows the overall process for multimodal data registration and visualization. We use color 3D point clouds as a common input format for 3D feature detection and matching because some inputs may not have mesh connectivity information. 3D data from 3D sensors or proxy computer graphics (CG) objects are directly registered and 2D data are registered via 3D reconstruction techniques such as stereo matching or SfM. In 3D reconstruction, camera poses are extracted so that the original capture locations and orientations can be simultaneously transformed in registration.

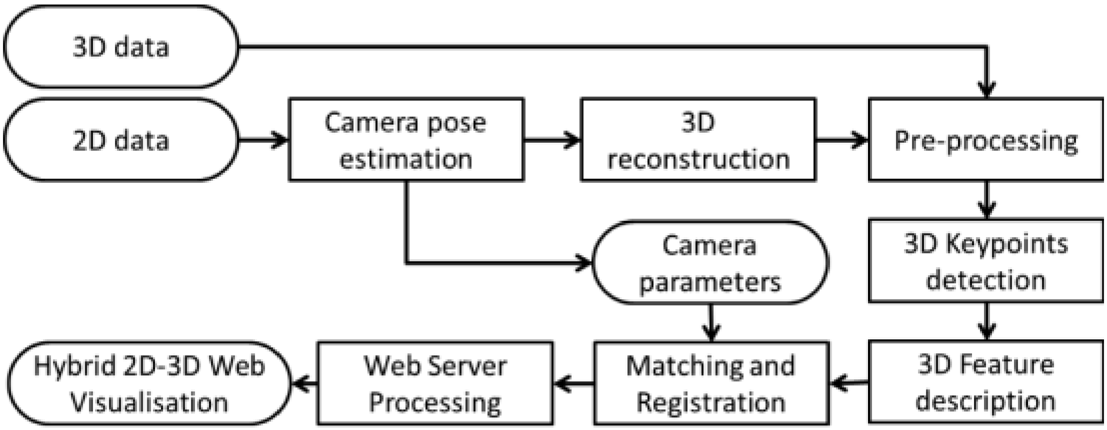


Fig. 2.
Pipeline for multimodal data registration and visualization.

Point clouds from different modalities have different density, and some of them have irregular sample distribution even in the same scene. For example, point clouds from an LIDAR scanner or spherical images become sparser as the distance from the capture device increases. This may cause bias in feature detection and description. We apply a 3D voxel grid filter which samples vertices in a uniform 3D grid to make the density of point clouds relatively even.

Keypoints are detected by the combination of a 3D Kanade-Tomasi detector [18] and 3D SIFT detector [33] (Section V-A). Then, multidomain 3D features are extracted in local, keypoint, and color domain as a 2D vector for each keypoint (Section V-B). The extracted feature descriptors from different modalities are matched to find the optimized registration matrix to the target coordinate system (Section V-C). The point cloud registration is refined over the whole point cloud using the iterative closest point (ICP) algorithm [34].

The complete data set is then organized and processed into a representation suitable for transmission over the Web. Video files are compressed using the OGG/Theora codec, and thumbnails are created from all image and video files. 3D point cloud data are entered into an octree data structure, which are traversed breadth first to create a series of binary files, ready for progressive download to the client. The final visualization is a Web application engine that mixes both 2D video, 2D image, and 3D WebGL contexts to allow users to navigate through the scene in an interactive manner. The application is designed to work on handheld devices as well.

SECTION IV.

Input Modalities

We consider a wide range of 2D/3D and active/passive sensors commonly used in various fields.

A. Light Detection and Ranging Sensor

LIDAR is an active sensing device using a light pulse signal to acquire 3D scene geometry. It is one of the most accurate depth ranging devices but has the limitation that it retrieves only a point cloud set without color or connectivity. However, some recent LIDAR devices provide colored 3D structure by mapping photos simultaneously taken during the scan. We have verified that color information is useful in multimodal data registration in [15], so we use FARO Focus 3D X130[2] to obtain colored 3D point clouds in this paper. Multiple scans acquired from different viewpoints are manually registered and merged into a complete scene structure using markers in the scene and the software tool provided with FARO. We do not use our automatic registration method for this partial scan registration because this LIDAR model will be used as a ground-truth target reference in our evaluation.

B. Spherical Imaging

A spherical camera captures a full surrounding scene visible from the camera location. Omni directional imaging is useful for environmental texture map generation or lighting source detection, but it always requires postprocessing to map the image in spherical coordinates to other images captured in a different coordinate system [35]. We assume that the scene is captured as vertical stereo pairs to allow dense reconstruction of the surrounding scene for automatic registration. We use Spheron,[3] a spherical line scan camera, and follow the stereo matching and reconstruction approach in [36].

C. Photographs

Digital photographs are the most common source of scene information. 3D reconstruction and camera pose estimation from multiview images has been actively researched for a long time. A set of photographs can be registered to a 3D space by registering the reconstructed 3D model because the camera poses are computed during the reconstruction process.
Bundler [37] followed by PMVS [38] provide a dense 3D reconstruction with camera pose estimation from multiple photos. Autodesk also provides an on-line image-based 3D reconstruction tool, RECAP360.[4] Both tools are used in our experiment.

D. 2D Videos

If a single moving video camera is used, the same approach in Section IV-C is used because video frames from a moving camera can be considered as multiview images. In case of multiple wide-baseline witness cameras, it is difficult to get the scene geometry for automatic registration if the camera viewpoints do not have sufficient overlap. In this paper, we define 2D videos as wide-baseline fixed witness cameras capturing a common space. Camera poses are estimated by wand-based calibration [39] aligned to the origin of the LIDAR sensor.

E. RGBD Video

Consumer level low-cost RGB+Depth cameras are becoming increasingly popular. Though infrared (IR) interference limits their validity in outdoor environments, they are still useful in indoor or shaded outdoor areas. KinectFusion [40] reconstructs a voxel volume from an RGBD video sequence by camera pose estimation and tracking. We use the Xtion PRO camera[5] to acquire an RGBD video stream of the scene.

F. Proxy Model

Proxy model means a simple CG object that represents or symbolizes real 3D objects. Proxy models are used in areas such as augmented reality, previsualization, virtual maps, and urban planning. They are normally generated by CG, but there are some semi/fully automatic algorithms such as plane-/block-based scene reconstruction from images [41], [42]. SketchUp[6] provides a semiautomatic reconstruction using vanishing points alignment. It is useful to build simple scenes but takes a long time for complex scenes. We use an axis-aligned plane-based scene reconstruction from spherical images [43] in the experiments. In feature detection and description, the plane structure is densely sampled to extract sufficient points for feature computation.

SECTION V.

Multimodal Data Registration

A. 3D Feature Detector

Keypoint detection is an essential step prior to matching and registration. There are many 3D feature detection methods developed and evaluated [16], [17]. However, all detectors were evaluated for accurate 3D models generated by CG or single-modal sensors. Highly ranked detectors in those evaluations do not guarantee such high repeatability and distinctiveness for multimodal data sets that have potentially different types of errors, sampling characteristics and distortions. For example, heat kernel signature detector [44] shows good repeatability and distinctiveness in those evaluations, but is too selective to yield a sufficient number of repeatable keypoints between cross-modalities due to geometrical errors induced from incomplete 3D reconstruction methods. A feature detector which produces a relatively large number of evenly distributed keypoints is preferred for robust multimodal data registration. We consider color as well as geometry to extract the most information from input data sets with outliers and different sampling resolutions. We use the combination of 3D Kanade-Tomasi detector and 3D SIFT feature detector.

The original 2D Kanade-Tomasi detector [18] uses an eigenvalue decomposition of the covariance matrix of the image gradients. In the 3D version of the Kanade-Tomasi detector, 3D surface normal vectors calculated in the volume radius of rs are used as input. Eigenvalues represent the principal surface directions and the ratios of eigenvalues are used to detect 3D corners in the point cloud.

The SIFT feature detector [33] uses a Difference-of-Gaussian filter to select scale-space extrema then refines the results by Hessian eigenvalue test to eliminate low contrast points and edge points. We use 3D versions of the Kanade-Tomasi detector and the SIFT detector implemented in the open source Point Cloud Library.[7] Parameters for 3D SIFT feature detector are defined as [Minimum scale S_m , Number of octaves S_o , Number of scales S_s].

B. Multidomain Feature Descriptor

Most 3D feature descriptors rely only on local geometric or color features. However, these descriptors are not suitable for multimodal data registration because input sources may have a high level of geometric reconstruction error or different color histograms. Our preliminary research [15] found that the combination of descriptors applied on different domains such as color and geometry can improve the matching and registration performance for multimodal data.

We use the FPFH descriptor as a base descriptor because it shows fast and stable performance in our preliminary research [14]. FPFH uses a cumulated Simplified Point Feature Histogram (SPFH) [22]. SPFH extracts a set of tuples [α , φ , θ] from a keypoint p and its neighboring local points $\{p_k\}$, where α is angle to the second axis, φ is an angle to the first axis, and θ is a rotation on the UW plane. For neighboring local points, their k -nearest neighbors (k -NNs) are determined and the FPFH histogram is computed by weighted sum of their neighboring SPFH values as (1). The weight ω_k is a distance between points p and p_k . The number of bins is set as 11 for each α , φ , θ . Therefore one FPFH descriptor can be represented as a vector with 33 bins

$$\text{FPFH}(p)=\text{SPFH}(p)+\frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k}\cdot\text{SPFH}(p_k).\quad(1)$$

Source:

```
\begin{equation} \text {FPFH}(p) = \text {SPFH}(p) + {\frac{1
}{ k}} \sum _{i=1}^{k} {\frac{1 }{ \omega _{k}} \cdot \text
{SPFH}(p_{k} )}. \end{equation}
```

In this research, the FPFH descriptor is extended to multiple domains in order to utilize geometry and color information together. For the same input point cloud with detected keypoints, three different FPFH descriptors are calculated in three different domains: Local, Keypoint, and Color. The result is represented as a 2D vector with 33×3 bins.

FPFH in the local domain F_L defines the characteristic of local geometry calculated from a keypoint and its neighboring local 3D points in the volume radius of r_l as normal local descriptors. FPFH in the keypoint domain F_K defines the spatial distribution of detected keypoints, which represents semiglobal geometric feature of the scene. F_K is calculated from a keypoint and its neighboring keypoints in the volume radius of r_k , which is much larger than r_l . Finally, FPFH in the color domain F_C defines the color characteristics of a keypoint and its neighboring local 3D points in the same volume radius of r_l as F_L . F_C is calculated in the same way but uses color components instead of surface normal components. We use the CIELab color space which is more perceptually uniform than the RGB space as proved in [25].

C. Hybrid Feature Matching and Registration

We propose the Hybrid RANSAC registration method to find an optimal 3D rigid transform matrix between feature sets. This extends the SAC-IA algorithm [22] by introducing a new distance measure with weighted sum of multidomain FPFH descriptors. Fig. 3 presents a block diagram of the proposed feature matching and registration method for the registration of keypoint set P in the source model to keypoint set Q in the target model.

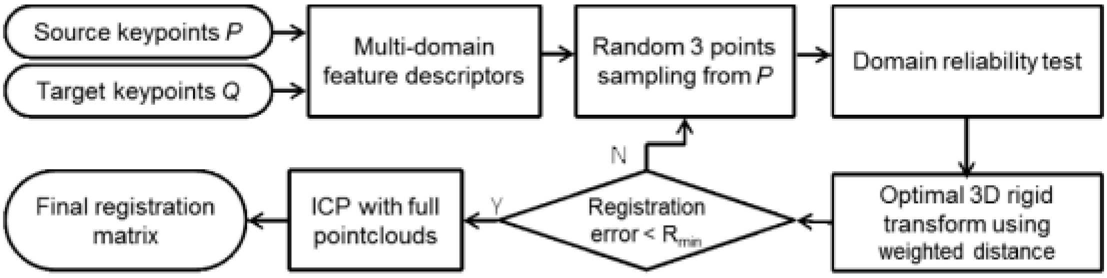

Source keypoints P
Target keypoints Q
Multi-domain feature descriptors
Random 3 points sampling from P
Domain reliability test
N
Optimal 3D rigid transform using weighted distance
Registration error < R_{min}
Y
ICP with full pointclouds
Final registration matrix

Fig. 3.
Hybrid RANSAC-based feature matching and registration.

The contribution of description domains in matching is adaptively selected according to the distinctiveness of the descriptor. If the point is selected from repetitive geometry or color patterns, it has a high possibility of a wrong match even with a low matching cost. The reliability $\lambda(p)$ for a point p is computed by the ratio of the second to first nearest neighbor distances in Q as shown in (2), where $D(p,q)$ denotes the distance between descriptors of p and q, and $p_{NN[\cdot]}$ an element of p's k-NN in Q

$$\lambda = D(p, p_{NN[1]}) / D(p, p_{NN[0]}). (2)$$

Source
```
\begin{equation} \lambda = D(p, p_{NN[{1}]}) / D(p,
p_{NN[{0}]}). \end{equation}
```

The total matching cost $D_T(p,q)$ for a source keypoint p to a target keypoint q with multiple domains is calculated by the weighted sum of individual domain descriptors as

$$D_T(p,q) = \lambda_L D_L(p,q) + \lambda_K D_K(p,q) + \lambda_C D_C(p,q). (3)$$

Source
```
\begin{equation} D_{T}(p,q) = {\lambda _{L}}D_{L}(p,q)+
{\lambda _{K}}D_{K}(p,q)\!+\! {\lambda _{C}}D_{C}(p,q).\qquad
\end{equation}
```

Algorithm 1 shows the registration process in detail.

Algorithm 1 Hybrid RANSAC Registration

Input:

Keypoint descriptor sets $P=\{p_i\}$ and $Q=\{q_i\}$

1. Ramdomly select 3 samples $S=\{s_i\} \subset P$

```
2. Calculate reliability set $\lambda(s_i)=\{\lambda_L(s_i),\lambda_K(s_i),\lambda_C(s_i)\}$ for $s_i$
3. Find matches $Q_s=\{q_{s(i)}\}\subset Q$ with $\min(D_T(s_i,q_{s(i)}))$
4. Compute a rigid 3D transform matrix $T$ from $S$ to $Q_s$
5. Exclude unreferenced keypoints in $T(P)$ and $Q$ which have no corresponding points within a range of $R_{max}$ from the keypoints
6. Compute registration error $E_R$ for the rest of keypoints in $T(P)$ and $Q$
7. If $E_R<E_{min}$ , then replace $E_{min}$ with $E_R$ and keep $T$ as $T_{opt}$
8. Repeat step 1–7 until it meets the termination criteria:
(a) Reach the maximum iteration $I_{max}$
(b) Emin<Rmin
Output: rigid 3D transform matrix $T_{opt}$
```

SECTION VI.

Web-Based Visualization

The Web-based visualization is based on a hybrid 2D–3D approach, mixing video, image, text, and 3D displays. The input data, while registered to a common 3D space, require preprocessing in order to ensure its suitability for transfer to, and rendering on, a remote Web client. These preprocessing steps and the rendering approaches used are discussed below.

The large amount of data with which we are dealing can make it difficult to enforce strict rules on file and directory structure and organization—real world big data are messy. Instead, we use a simple JSON file to store a scene description with relative paths to the location of the relevant data. The data are stored on a Linux-based machine running a custom Apache2 Web-service, which is configured to enable HTTP gzip compression for all the file formats that serve the client (including the custom binary formats as described below). This enabling of gzip for all files provides a final compression step which is extremely fast, as it relies on a well-understood algorithm encoded at a low level in both the server and client-browser application, and thus adds very little processing overhead for potentially significant reductions in file size [30].

On the client side, the hybrid 2D–3D renderer is setup with a base WebGL 3D context running in an HTML5 canvas element, which is supplemented by various 2D document object model (DOM) elements, described below. Interactive scene navigation is controlled by rotating, panning, and zooming with standard mouse/touch gestures.

A. Progressive Point Cloud Rendering

The registered point clouds generated by the various input modalities presented in Section IV (from raw point cloud scans or reconstructed image data) are initially in off format, encoding position and color of each point. File sizes range from tens to hundreds of megabytes. Rendering such data in an offline context is trivial; doing so in a Web browser context, however, presents two principal challenges. The first is the simple time taken to download such data. Even with a fast Internet connection, and the colored points represented in binary format, and compressed using the HTTP standard gzip algorithm, it would take several seconds or even minutes to download the data before it can be rendered. Second, such a large number of points can easily overwhelm the browser application—our initial tests, on modern hardware, with a very simple WebGL point cloud rendering application showed that a maximum of 3.5M points can be rendered before the application crashes (in comparison, a similar offline application can render many more points).

Using a hierarchical data structure to store and transmit the data solves both of these issues. Not only does it permit lower resolution versions of the data set to be transmitted and rendered immediately, while further data are downloaded, but it also permits rendering of larger data sets which would not be possible to render at full resolution in the browser. Thus, we preprocess our data in a similar way to [30], organizing the data hierarchically into a memory efficient octree, where the center of each node is stored, along with the mean color of all the points stored within it and any of its child nodes. An offline process parses this octree breadth first and outputs the position and color information in a simple binary format, which is then stored in a sequence of files. Each file contains a maximum of 5000 entries, each entry corresponding either to point representing a node of the octree, or a point of the final data set.

The browser application features as its base context a WebGL rendering engine which downloads sequentially the file sequence described above. Each file is processed and the data for the points uploaded to the GPU. The resolution level (i.e., depth into the octree) is tracked, so that when higher resolution data are downloaded and displayed, lower resolution data are discarded (to avoid occlusion issues). The result is that, upon loading of the Web page, an initial low-resolution version of the point cloud is quickly displayed on the screen, which is then refined to a higher resolution version as more data are downloaded, until the final point-cloud is displayed.

Multiple point clouds can be downloaded and rendered simultaneously, and hidden/shown using a simple GUI element. This capability for visualization of multiple point clouds allows the user to quickly see the similarities and differences between the data obtained from the different modalities, which plays an important role in assessing the data quality, completeness, and key requirements.

B. Video Data and Timeline

The raw video footage recorded from the witness cameras (Section IV-D) is initially stored in uncompressed format. For transfer to the remote client it is compressed and reduced in resolution using the OGG-Theora codec at medium quality. A thumbnail image of a fixed frame from the first seconds of each video is also created. Upon loading the Web page, all videos are preloaded into the page DOM as HTML5 video elements, which are hidden from view using CSS (the elements are required to stream the video data from the server, but the actual frames will be rendered in WebGL as described below).

Witness cameras are represented in the 3D scene by simple plane meshes whose positions and orientations match those extracted as above. The video footage from each camera is then rendered in the 3D context, extracting the image data from the HTML5 video element and passing it as a WebGL texture, which is displayed on the relevant plane mesh for each camera (Fig. 14). This extraction of video frames from the HTML5 video element for use as textures within a 3D context is one of the major benefits of developing a hybrid interface within a Web-based context, as such a pipeline in a standard desktop OpenGL context requires a greater level of software engineering and preprocessing [45].

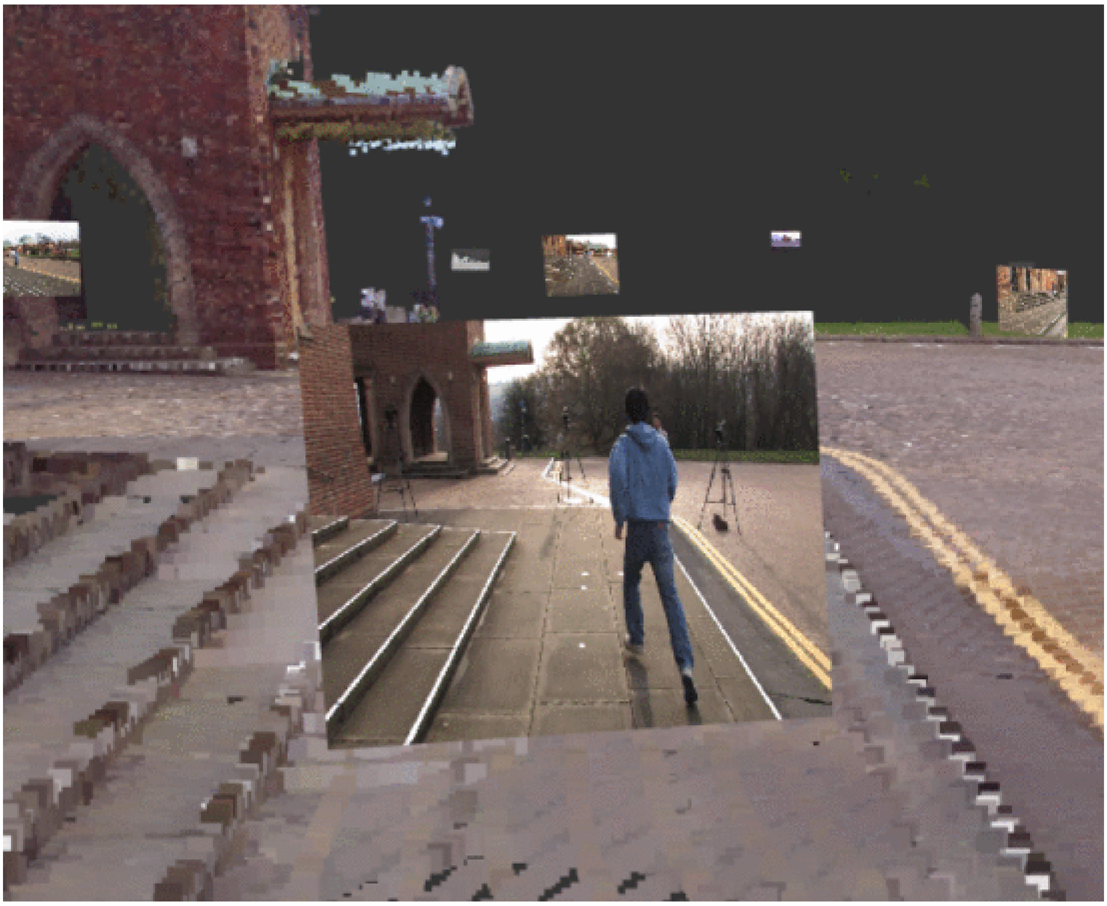
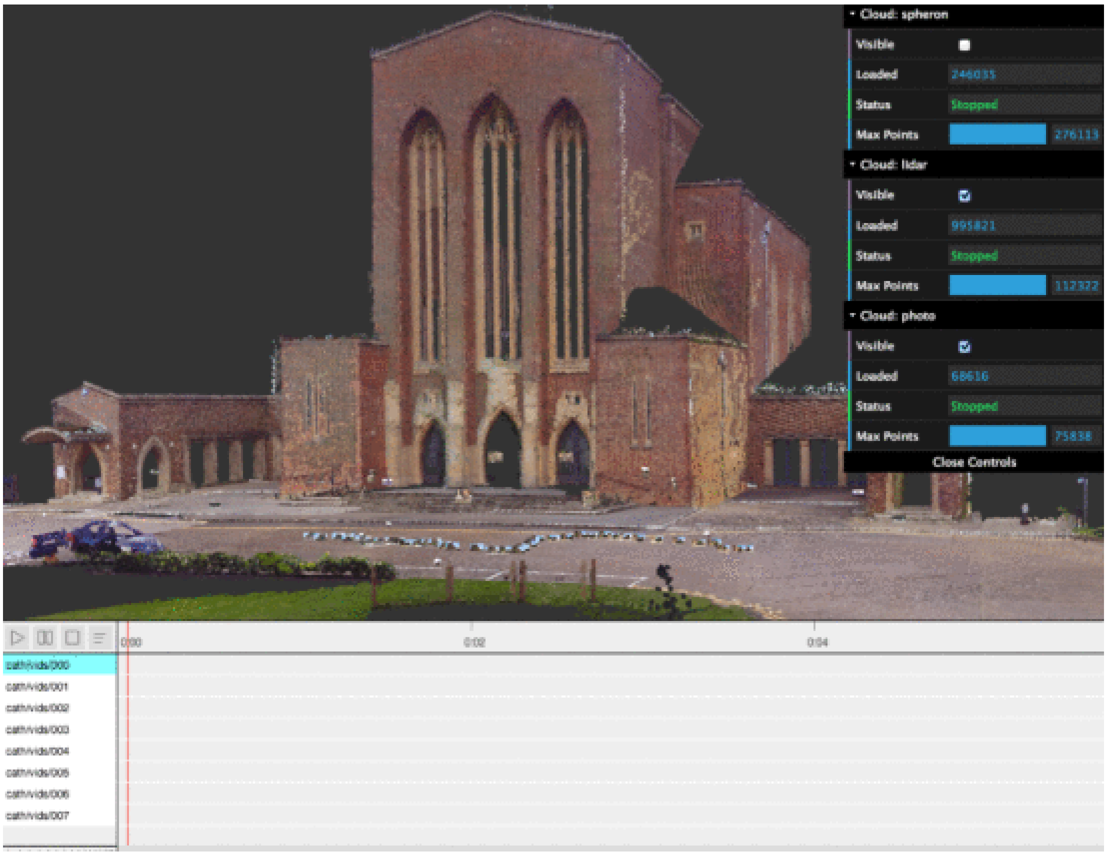


Fig. 14.
(Hidden) HTML5 video element pipes texture information, at 30 frames/s, positioned to the original camera location and orientation.

To control playback, position, and scrubbing, a simple timeline interface is drawn in a 2D canvas (Fig. 15). The timeline allows selection of which video to play, along with playback controls and a draggable timeline bar to control scrubbing. Video buffering is used to ensure that enough video data have been downloaded to pass as texture information to the WebGL renderer, and also to ensure the scrubbing interface is synchronized to the video footage. Upon selecting a witness camera in the timeline interface, the camera position in the 3D scene is instantly moved to a position just behind the plane mesh representing that camera, allowing the video to be seen within the 3D context. For performance reasons, only one video can be played at a time (the video which is selected in the timeline interface).



▾ Cloud: spheron
Visible □
Loaded 246035
Status Stopped
Max Points 276113

▾ Cloud: lidar
Visible ☑
Loaded 995821
Status Stopped
Max Points 112322

▾ Cloud: photo
Visible ☑
Loaded 68616
Status Stopped
Max Points 75838

Close Controls

Fig. 15.
Hybrid 2D–3D Web interface showing the timeline component and GUI overlaying the 3D context.

C. Sensor Raw Data Billboards

The registration process described in Section V also outputs the positions of the various sensors (LIDAR, Spheron, RGBD camera, and regular photo cameras), which are registered to the combined LIDAR scan for reference. To visualize these sensor positions, we render a simple mesh plane at the 3D position of the sensor within the scene, and pass a thumbnail image of the original sensor image as a texture for that plane. Unlike the similar setup for witness cameras, for the sensors, we strip all rotational information out of the Model-View-Projection matrix immediately prior to rendering. This means that the plane meshes act as billboards, constantly rotating to face the camera, to best show the original sensor data. When the user clicks (or touches) the screen, a ray is fired into the scene and a simple collision detection algorithm determines whether the user has clicked on a billboard or not. If so, the 3D context is faded into the background and the original, full-resolution image of the sensor is shown in an HTML/CSS lightbox (Fig. 16).

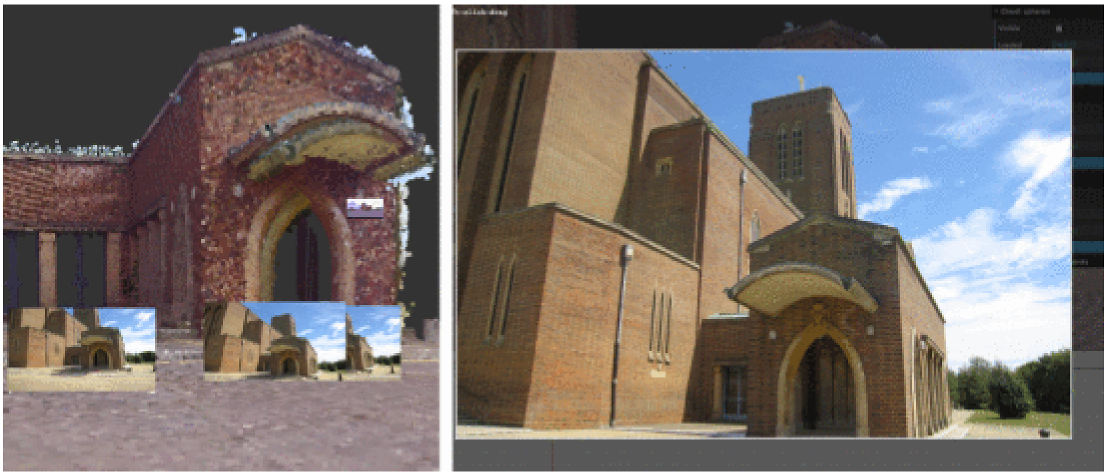


Fig. 16.
Left: sensors represented as billboards with a thumbnail of original image. Right: clicking on the billboard displays the full resolution image.

The billboards can occasionally be difficult to spot among the rest of the point cloud data, so we have added a feature where the user can enable an interface overlay which draws colored lines above each billboard, thus highlighting the locations of all the sensors. Different sensor types can be assigned different colors.

D. Annotation Component

One of the potential industrial benefits of the system presented in this paper is that it permits various professional users to view and interact with the same data, at the same time, while potentially being in different physical locations. The rise of remote collaborative working, seen most strongly with the popularity of online tools such as Google Docs and Dropbox, has yet to reach the 3D production and post-production world, largely due to problems which the work in this paper strives to overcome.

While a full collaborative work application lies as a potential future goal, we have implemented an annotation component, which permits users to annotate areas of the data set, raising the possibility of those annotations being stored on a server for viewing by other users. Annotation of point cloud data is slightly more troublesome than when dealing with mesh data. In the latter case, a simple raycast-mesh collision detection is enough to detect the 3D point where the user has clicked (or tapped) on the scene. GL points, however, are drawn as pixels and do not have any representative volume, thus a simple raycasting method is not sufficient. To counter this problem, we recreate in the browser context the octree used for the initial data partitioning, and calculate ray collisions on the nodes of the octree. This permits us to effectively discover the 3D point in the scene with which the user has interacted, and allows us to associate (and draw) an annotation at that point (Fig. 17).

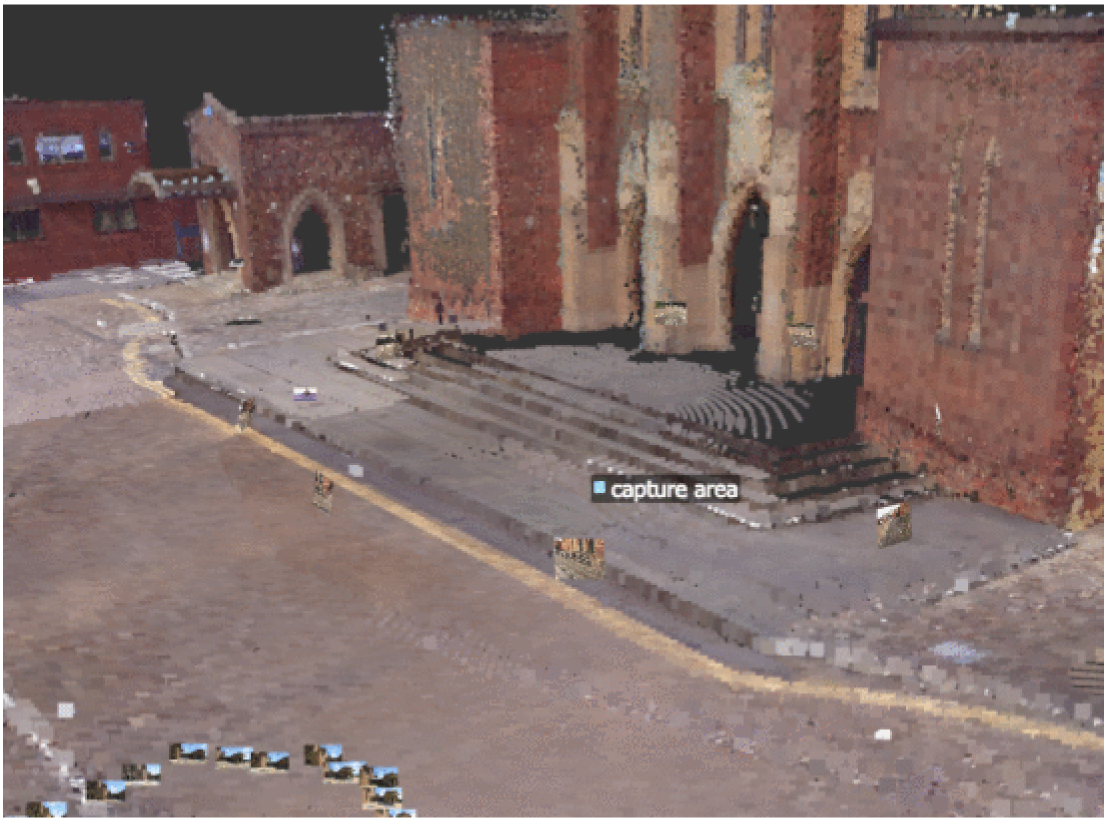


Fig. 17.
Screenshot showing an example of the annotation component being used to label elements in the scene.

SECTION VII.

Public Multimodal Database

To support research into multimodal data processing, we present a big multimodal database acquired in various indoor and outdoor environments, available at: http://cvssp.org/impart/.

The database includes raw capture data and 3D reconstructions for various indoor/outdoor static scenes and multiple synchronized video captures for dynamic actions in the scene. Various capture devices such as gray/color LIDAR scanners, spherical camera, DSLR/compact still cameras, HD (1920×1080) video cameras, HD 2.7 K/4 K cameras, and RGBD cameras were used. The HD video cameras were genlock synchronized and calibrated. The repository contains detailed notes on the capture, and some preprocessing is available to make the data set more useful to researchers. Details can be found in the capture notes provided on the repository [46].

The proposed registration and visualization pipeline is tested on three data sets from this repository: Studio, Patio, and Cathedral. The Studio set is an indoor scene with stable lighting condition provided by KinoFlo fluorescent lights on the ceiling. The Patio set is an outdoor scene covering around 15 m $\times10$ m area. The main capture area is surrounded by walls, has a symmetric structure and includes repetitive geometry and texture patterns from bricks and windows. RGBD data can be acquired for this scene without IR interference because it is shaded area. Fifteen HD video cameras were used to record main actions in the scene. The Cathedral set is a large outdoor scene covering around 30 m ×20 m open area. The scene was captured under the direct sun light which resulted in changing brightness and shadows. Main actions were recorded by eight HD video cameras. Fig. 4 shows examples of static and dynamic captures for the test scenes. As mentioned in Section IV-D, multiple HD video cameras are registered using their extrinsic camera parameters in our experiments because the cameras are too sparsely placed (little overlap) to recover the background geometry from dynamic videos.

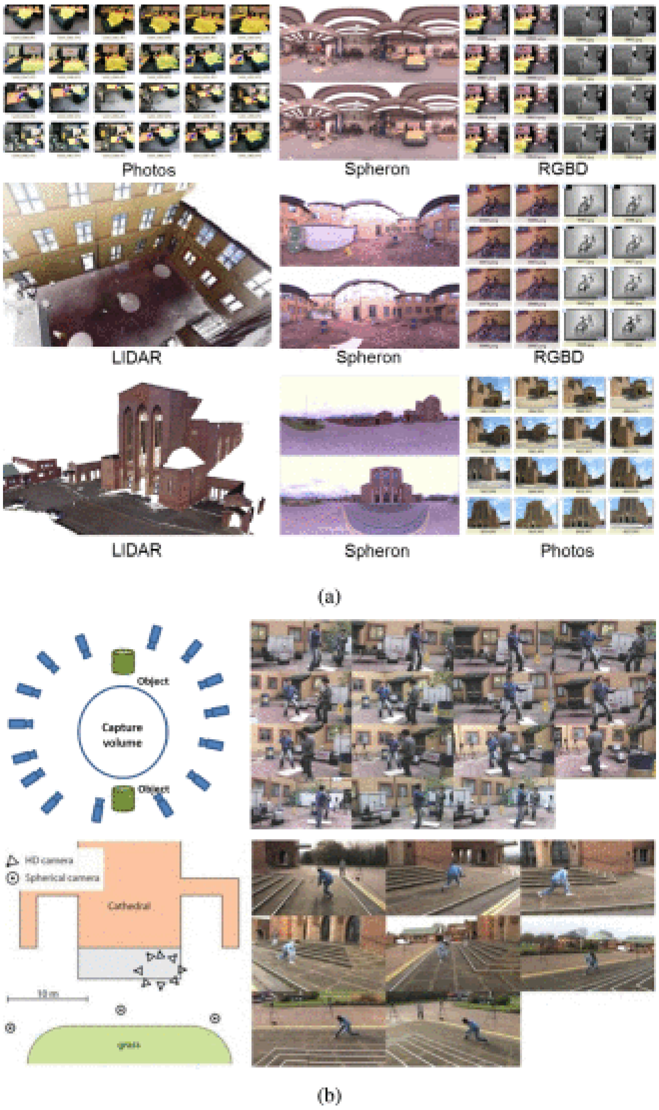


Fig. 4.

Examples of multimodal data sets. (a) Static scene capture (top: Studio, middle: Patio, bottom: Cathedral). (b) Dynamic scene capture (top: Patio, bottom: Cathedral).

SECTION VIII.

Experiments

In order to evaluate general performance of the proposed multidomain feature descriptor and hybrid-registration to single modality cases, we tested them on the RGB-D Scenes data set from University of Washington.[8] It provides 3D color point clouds of four indoor scenes. Each scene has 3–4 takes with different main objects and coverage for the same background scene. We randomly merged the first takes of each scene into one model as shown in Fig. 5(a), and tried to register the second takes of each scene in Fig. 5(b) to the merged target scene in Fig. 5(a). Different objects and coverage of the second takes can be considered as noise or errors against the target scene, which makes the test more challenging. For objective evaluation, we generated a ground-truth registration by manual 4-points matching and ICP refinement using MeshLab.[9]

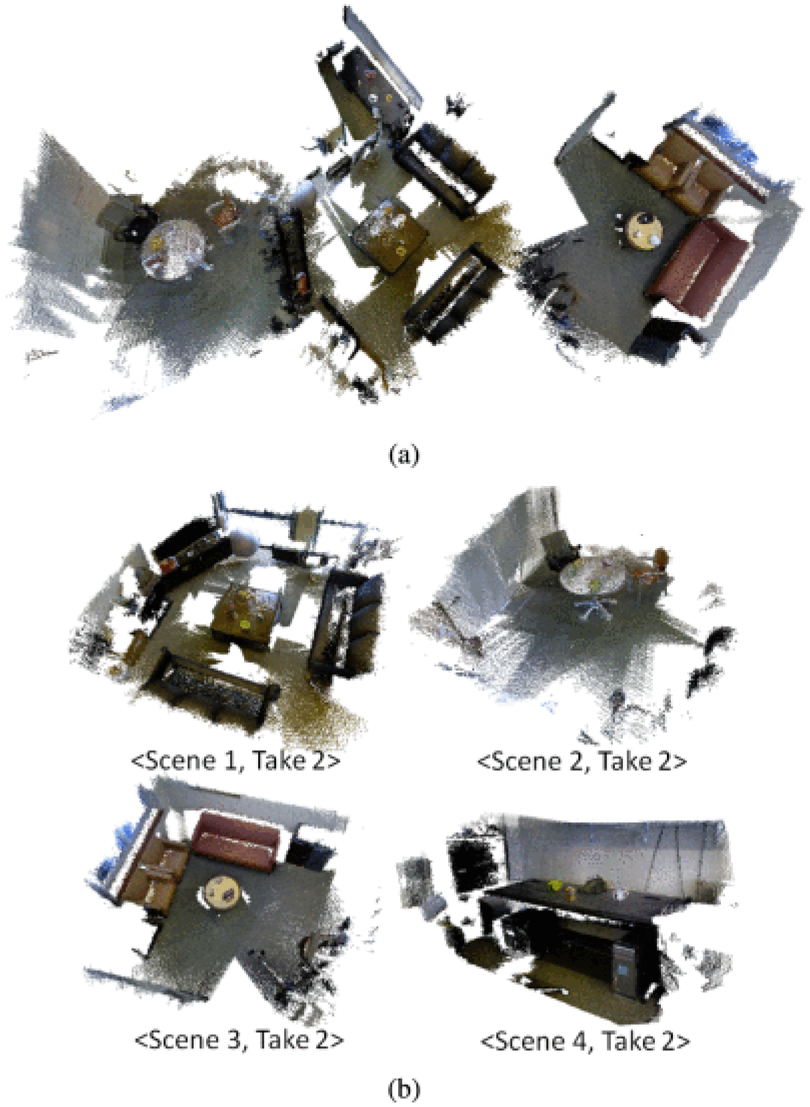
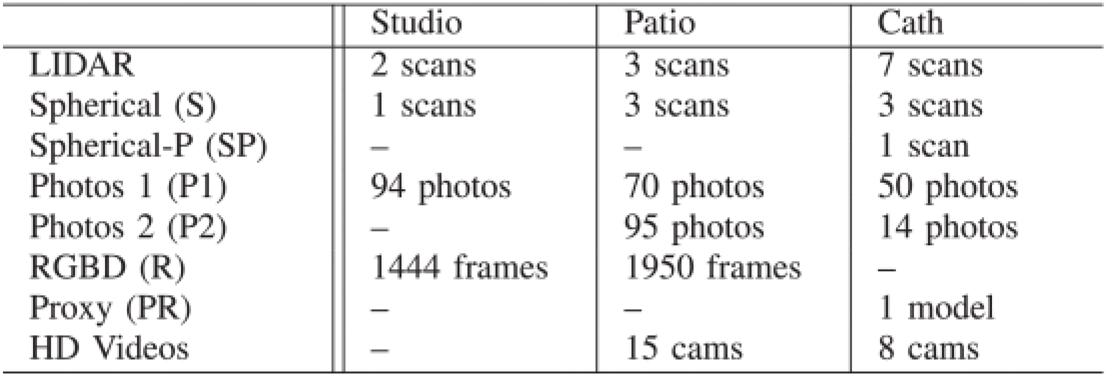


Fig. 5.
Washington RGB-D scenes data set. (a) Target reference from takes 1. (b) Test sets to be registered.

In the experiments on the multimodal data sets introduced in Section VII, the LIDAR scan in each scene is set as the target reference and all other models are registered to the LIDAR coordinate system. Table II shows the data sets used in the experiments. “Spherical-P” is a partial spherical reconstruction to verify the performance of part registration to the whole scene. 3D models are reconstructed for the real world scale using the reconstruction method introduced in Section IV. In reconstruction from photographs, Autodesk RECAP360 is used for the Studio and Patio scenes, and the Bundler [37] + PMVS [38] for the Cathedral scene to test various algorithms. HD videos are not tested for reconstruction and registration because they have been calibrated to the LIDAR coordinate system using the camera calibration process. The 3D point clouds reconstructed from 2D data for the experiments are illustrated in Fig. 6.

TABLE II Experimental Data Sets



	Studio	Patio	Cath
LIDAR	2 scans	3 scans	7 scans
Spherical (S)	1 scans	3 scans	3 scans
Spherical-P (SP)	–	–	1 scan
Photos 1 (P1)	94 photos	70 photos	50 photos
Photos 2 (P2)	–	95 photos	14 photos
RGBD (R)	1444 frames	1950 frames	–
Proxy (PR)	–	–	1 model
HD Videos	–	15 cams	8 cams

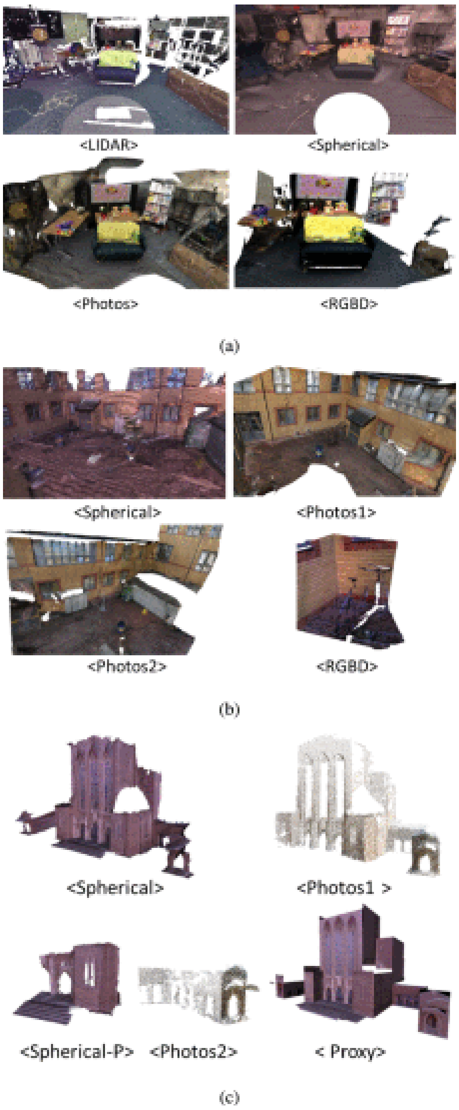


Fig. 6.

3D models for registration. (a) Studio. (b) Patio. (c) Cathedral.

Ground-truth registration was generated as the same manner as the Washington data set. Fig. 7 illustrates the original data sets, ground-truth registration results, and the registration error maps. The error map shows Hausdorff distance to the LIDAR model mapped in the range of 0–3 m to a Blue-Red color range. We observe that even the ground-truth registration has errors against the target model because the source model has reconstruction errors, different coverage and density. Therefore, we measure the rms error to the ground-truth registration points instead of the distance to the LIDAR model for the registration evaluation.

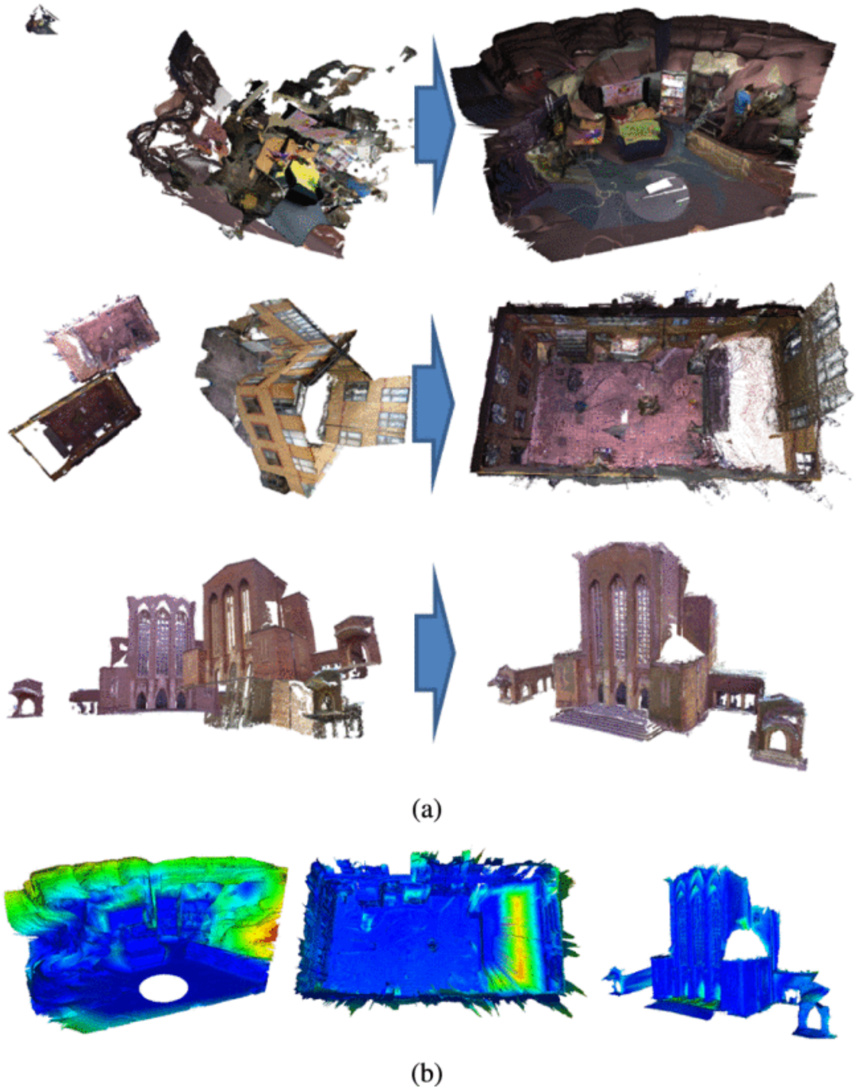


Fig. 7.
Ground-truth registration. (a) Registration (top: Studio, middle: Patio, bottom: Cathedral). (b) Error map of spherical model (left: Studio, middle: Patio, right: Cathedral).

In 3D point cloud registration, the ICP algorithm requires an initial alignment. It fails in registration if the initial position is not close enough to the final position. Therefore, we judge the performance of initial registration by success or failure of the following ICP refinement. We found that the ICP converges successfully if the initial registration is within 1–2 m of rms error range to the ground truth registration.

A. 3D Feature Detector

In this experiment, we evaluate existing 3D feature detectors, and then analyze their influence on the registration performance. We test three feature detectors and their combinations: 3D Noble [47], 3D SIFT, 3D Tomasi, 3D Noble+SIFT, and 3D Tomasi+SIFT. We do not test the combination of 3D Noble and Tomasi because both are geometry-based detectors. Testing is performed on our multimodal data set. The range parameter r_s for surface normal calculation is set as 0.5 and 0.2 m for the outdoors scenes and indoor scene, respectively. The scale parameters for the SIFT detector are set as $[S_m, S_o, S_s]$ = $[r_s$, 8, 10] as suggested in the original implementation.

Detected keypoints for the spherical reconstruction of the Cathedral scene are shown in Fig. 8. The Noble detector detected four times more points than other detectors but they are concentrated in specific regions. The SIFT and Tomasi detectors detected similar number of feature points but the result of Tomasi is more evenly spread.

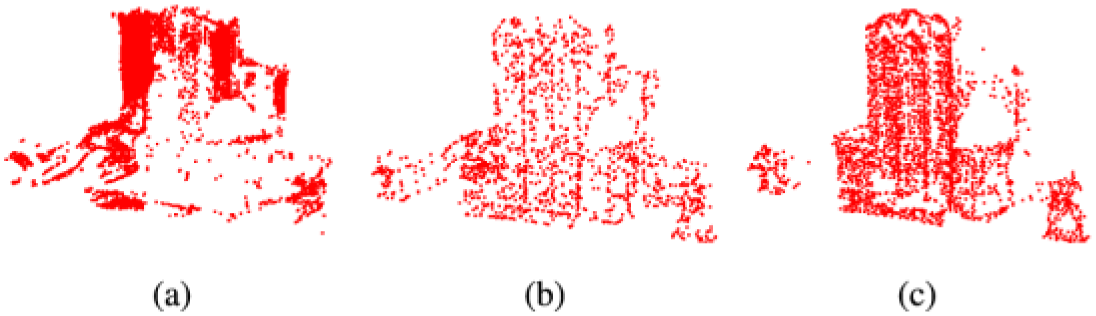
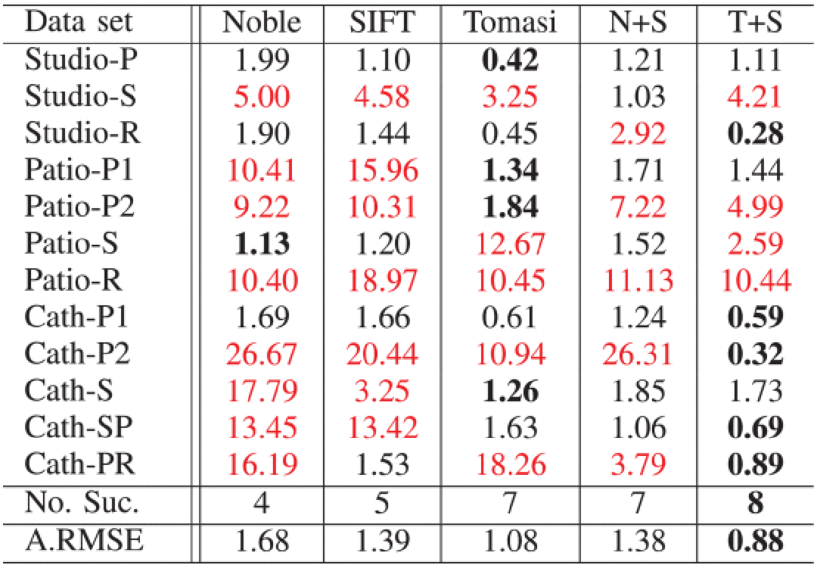


Fig. 8.
Feature detection result (Cath-S). (a) Noble (9729 points). (b) SIFT (2115 points). (c) Tomasi (2461 points).

The registration result using the detected keypoints in Table III clearly shows the influence of the feature detectors to matching and registration. In feature description and matching, we used the local FPFH descriptor with the parameter set $[r_l, R_{min}, R_{max}, I_{max}]$ = [0.8(outdoor)/0.3(indoor), 0.2, 0.8, 8000] in an intuitive way considering the scale of the scenes. They are fixed for all multimodal data sets because they are not sensitive to the scene scale or characteristics across the range from small scale indoor scenes to large scale building exteriors such as the Cathedral. Different parameters have been used only for the Washington data sets because their scale is unknown. In Table III, figures colored in red show failed cases in initial registration and bold ones show the best. $No.Suc$. means the number of models succeeded in initial registration for ICP, and $A.RMSE$ means the average rms registration error of the successful registrations. The Noble detector shows the worst performance in the single detector test in spite of the largest number of feature points because the points gathered in specific areas do not contribute to efficient matching and registration. The Tomasi detector shows the best performance among the single detectors with the largest number of successful registrations and the lowest rms registration error. The combinations of geometric and color detectors show better results as expected. Especially, the Tomasi+SIFT detector shows good registration performance even with a normal FPFH descriptor though it still fails with the Patio set due to its repetitive geometry and texture. We use this Tomasi+SIFT detector for multidomain feature description and Hybrid matching in the next section.

TABLE III Registration Result With Different Feature Detectors (N+S: Noble+SIFT, T+S: Tomsi+SIFT, S: Success, and F: Failure)



Data set	Noble	SIFT	Tomasi	N+S	T+S
Studio-P	1.99	1.10	**0.42**	1.21	1.11
Studio-S	5.00	4.58	3.25	1.03	4.21
Studio-R	1.90	1.44	0.45	2.92	**0.28**
Patio-P1	10.41	15.96	**1.34**	1.71	1.44
Patio-P2	9.22	10.31	**1.84**	7.22	4.99
Patio-S	**1.13**	1.20	12.67	1.52	2.59
Patio-R	10.40	18.97	10.45	11.13	10.44
Cath-P1	1.69	1.66	0.61	1.24	**0.59**
Cath-P2	26.67	20.44	10.94	26.31	**0.32**
Cath-S	17.79	3.25	**1.26**	1.85	1.73
Cath-SP	13.45	13.42	1.63	1.06	**0.69**
Cath-PR	16.19	1.53	18.26	3.79	**0.89**
No. Suc.	4	5	7	7	**8**
A.RMSE	1.68	1.39	1.08	1.38	**0.88**

B. Feature Matching and Registration

3D feature descriptors are computed for the keypoints extracted by the combination of Tomasi and SIFT in Section VIII. A . We compared the registration performance of the proposed multidomain FPFH descriptor and Hybrid RANSAC registration (denoted as F_{HYB}) with those of normal FPFH (F), SHOT (S), and cascade combinations of FPFH descriptors in different domains (F_{LK} , F_{LC} , and F_{LKC}). We use the same parameter set of Section VIII. A for the multimodal data sets and [r_l , r_k , R_{min} , R_{max} , I_{max}] = [0.2, 1.0, 0.05, 1.0, 5000] for the Washington data sets.

For matching performance evaluation, best matching pairs of all detected keypoints to the target reference are calculated and compared with the ground-truth feature matching pairs. Ground-truth feature matching pairs are defined by the closed keypoints of the target reference in the range of r_{gt} from the source keypoints transformed by the ground truth registration. r_{gt} was set as 0.03 for the Washington data set (the scale of the 3D coordinate is unknown) and 5 cm for the multimodal data set. As tested in [21], precision values are computed as follows:

$$\text{Precision}=\text{Number of correct matches}\text{Number of matches}.(4)$$

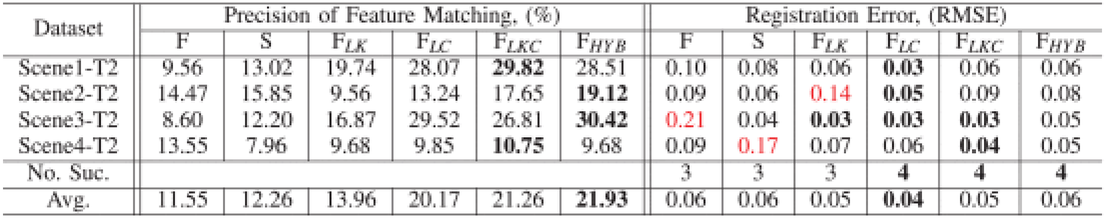
Source

```
\begin{equation} \text {Precision} = \frac {\text {Number of correct matches}}{\text {Number of matches}}. \end{equation}
```

1) Test on Single-Modal Data Set:

Table IV shows matching precision and registration results of the Washington RGB-D scenes data set according to the description methods. Only precision results are given here because the outlier ratio is more important in RANSAC-based registration. Average in the last row means the average of the whole precision values in the precision columns and the average rms registration error of the "successful registrations" in the registration columns. In the feature matching evaluation, combination of features from various domains shows higher precision rate. Especially, it shows better results both in feature matching and registration when the color information was involved because their appearance was captured with the same lighting condition and sensor. Feature matching shows relatively high precision rate though they were captured with slightly different objects and coverages. The proposed multidomain feature description and hybrid RANSAC registration shows competitive performances against other cascade combination methods but is not very advantageous considering its computational complexity.

TABLE IV Matching and Registration Results for Single-Modal Data Set (Washington Data Set)

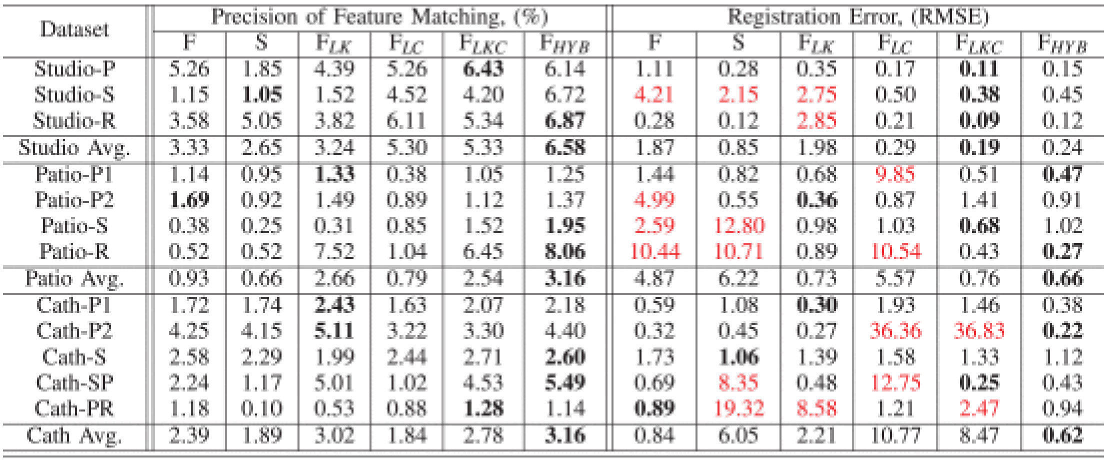


Dataset	Precision of Feature Matching, (%)						Registration Error, (RMSE)					
	F	S	F_{LK}	F_{LC}	F_{LKC}	F_{HYB}	F	S	F_{LK}	F_{LC}	F_{LKC}	F_{HYB}
Scene1-T2	9.56	13.02	19.74	28.07	**29.82**	28.51	0.10	0.08	0.06	**0.03**	0.06	0.06
Scene2-T2	14.47	15.85	9.56	13.24	17.65	**19.12**	0.09	0.06	0.14	**0.05**	0.09	0.08
Scene3-T2	8.60	12.20	16.87	29.52	26.81	**30.42**	0.21	0.04	**0.03**	**0.03**	**0.03**	0.05
Scene4-T2	13.55	7.96	9.68	9.85	**10.75**	9.68	0.09	0.17	0.07	0.06	**0.04**	0.05
No. Suc.							3	3	3	**4**	**4**	**4**
Avg.	11.55	12.26	13.96	20.17	21.26	**21.93**	0.06	0.06	0.05	**0.04**	0.05	0.06

2) Test on Multimodal Data Set:

Table V shows the feature matching and initial registration results of the multimodal data set. The precision rates of feature matching are much lower than those of single-modal set shown in Table IV due to different characteristics and reconstruction errors of modalities. The proposed feature description and matching method shows higher precision compared with other descriptions. Fig. 9 shows examples of feature matching according to descriptors. The best 20 keypoints matches for the Patio set and 200 matches for the Cathedral set using conventional SHOT and FPFH local descriptors and the proposed multidomain hybrid matching are visualized. The local descriptor matching results are scattered over the scene while the proposed method shows more consistent matching to the correct position.

TABLE V Matching and Registration Results for Multimodal Data Set



Dataset	Precision of Feature Matching, (%)						Registration Error, (RMSE)					
	F	S	F_{LK}	F_{LC}	F_{LKC}	F_{HYB}	F	S	F_{LK}	F_{LC}	F_{LKC}	F_{HYB}
Studio-P	5.26	1.85	4.39	5.26	**6.43**	6.14	1.11	0.28	0.35	0.17	**0.11**	0.15
Studio-S	1.15	**1.05**	1.52	4.52	4.20	6.72	4.21	2.15	2.75	0.50	**0.38**	0.45
Studio-R	3.58	5.05	3.82	6.11	5.34	**6.87**	0.28	0.12	2.85	0.21	**0.09**	0.12
Studio Avg.	3.33	2.65	3.24	5.30	5.33	**6.58**	1.87	0.85	1.98	0.29	**0.19**	0.24
Patio-P1	1.14	0.95	**1.33**	0.38	1.05	1.25	1.44	0.82	0.68	9.85	0.51	**0.47**
Patio-P2	**1.69**	0.92	1.49	0.89	1.12	1.37	4.99	0.55	**0.36**	0.87	1.41	0.91
Patio-S	0.38	0.25	0.31	0.85	1.52	**1.95**	2.59	12.80	0.98	1.03	**0.68**	1.02
Patio-R	0.52	0.52	7.52	1.04	6.45	**8.06**	10.44	10.71	0.89	10.54	0.43	**0.27**
Patio Avg.	0.93	0.66	2.66	0.79	2.54	**3.16**	4.87	6.22	0.73	5.57	0.76	**0.66**
Cath-P1	1.72	1.74	**2.43**	1.63	2.07	2.18	0.59	1.08	**0.30**	1.93	1.46	0.38
Cath-P2	4.25	4.15	**5.11**	3.22	3.30	4.40	0.32	0.45	0.27	36.36	36.83	**0.22**
Cath-S	2.58	2.29	1.99	2.44	2.71	**2.60**	1.73	**1.06**	1.39	1.58	1.33	1.12
Cath-SP	2.24	1.17	5.01	1.02	4.53	**5.49**	0.69	8.35	0.48	12.75	**0.25**	0.43
Cath-PR	1.18	0.10	0.53	0.88	**1.28**	1.14	**0.89**	19.32	8.58	1.21	2.47	0.94
Cath Avg.	2.39	1.89	3.02	1.84	2.78	**3.16**	0.84	6.05	2.21	10.77	8.47	**0.62**

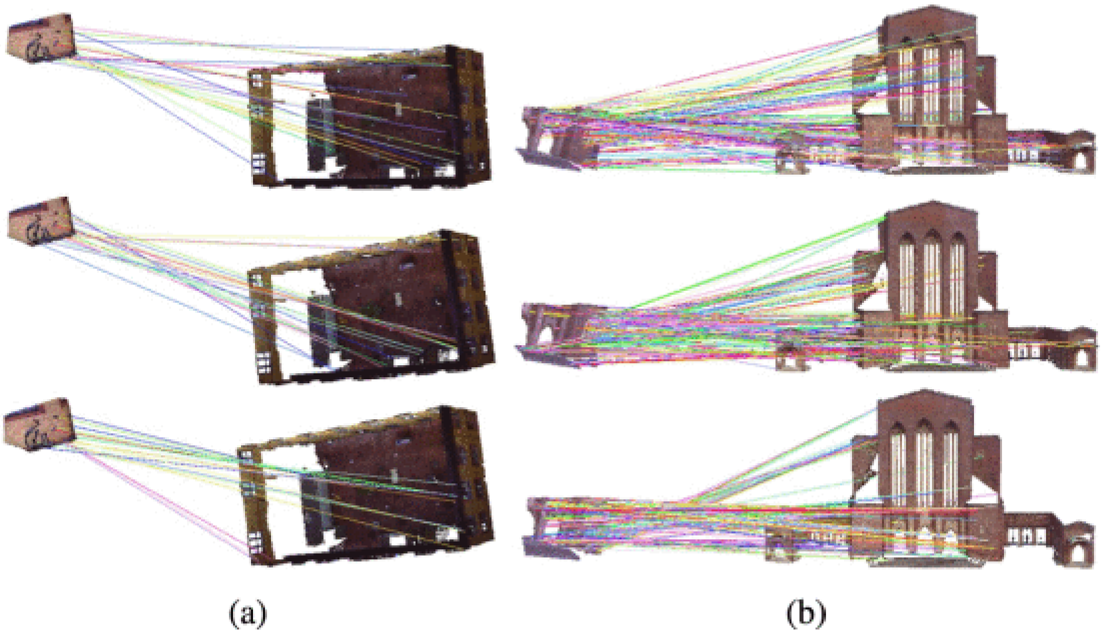


Fig. 9.

Matched features (top: SHOT, middle: FPFH, bottom: Proposed). (a) Patio-R to LIDAR. (b) Cath-SP to LIDAR.

In Table V, the Studio set shows better performance than Patio and Cathedral sets in matching and registration, and especially, the color information improves the performance of feature matching because the Studio set was captured in stable lighting condition. However, it shows poor result with the spherical reconstruction, because the Studio-S model was reconstructed from only one pair of spherical images and has large self-occlusion areas in the geometry.

The Patio scene models have repetitive structures with similar colors such as bricks and window frames. It causes relatively low feature matching rates compared with other data sets. In the registration results, we observe that some structures are misregistered by 180° as shown in Fig. 10(a). Keypoint descriptions (F_K) that consider feature distribution over a large area achieve better performance than local color or shape descriptors due to repetitive local geometry and appearance.

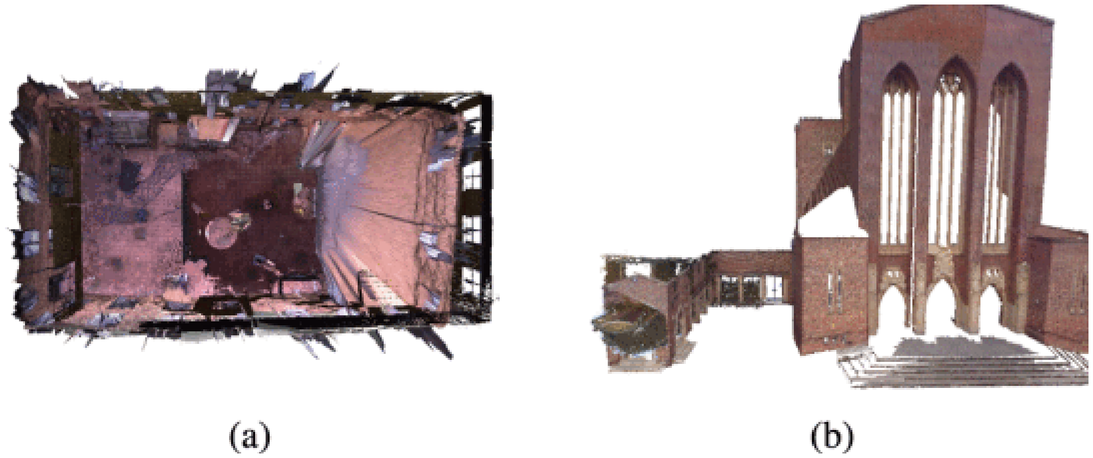


Fig. 10.
Failure cases in registration. (a) Patio-S with FPFH. (b) Cath-P2 with FPFH_{LC}.

In the Cathedral scene models, the appearance information is less trusted because it changes according to the capture device, capture location (direction) and time in the open outdoor environment. As shown in Fig. 10(b), the left wing of the building is mapped to the right wing in the LIDAR model. It happens with F , $\mathrm{F}\mathrm{LC}$, and $\mathrm{F}\mathrm{LKC}$ descriptors whose local and color components dominate the matching over the semiglobal geometric component. The proposed hybrid matching and registration sorts out this bias problem. However, the color information is more helpful than others in the case of proxy model (Cath-PR) whose distinctiveness of geometrical features are very low. SHOT descriptor also shows poor result in feature matching. This results from the failure of defining local reference frame for SHOT descriptor.

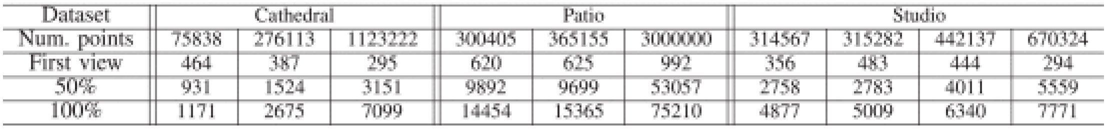
The cascade combinations of descriptors generally show slightly better performances than the single local descriptors, but it sometimes makes worse as seen in the case of Patio-P1 with $\mathrm{F}\mathrm{LC}$, Cath-P2 with $\mathrm{F}\mathrm{LK}$, Cath-SP with $\mathrm{F}\mathrm{LC}$ and Cath-PR with $\mathrm{F}\mathrm{LK}$. They show poor performances because the features from different domains compete each other without considering their reliabilities. The proposed matching and registration method FPFH_{HYB} successfully registered all 12 data sets with high precision feature matching and low rms registration error.

C. Web-Based Visualization

Figs. 11–17 show screenshots of the various components of the visualization. Table VI contains results showing the total time taken for the point cloud data (from all sources) to download and render, at clamped bandwidth of 8 Mb/s. The purpose of this table is to highlight the advantage of the LOD approach compared to simply waiting for the entire data set to download. Note that an initial, low resolution view is available within a second (note that the first view values are not related to the final size), yet the final data set (millions of points) may take several tens of seconds to download—without the progressive refinement technique, the user would be waiting approximately this time to see anything. These values are similar to those we obtained with the similar technique presented in [30], despite this there are multiple point clouds (at least three) being downloaded simultaneously, and compare well with those state of the art on the different but related problem of progressive mesh transmission [32].

The thumbnail images do not add overhead with respect to the 3D point cloud data, as their file sizes are comparatively small and they appear rapidly in the scene.

TABLE VI Time Taken (ms) to Download and Render Different Point Clouds at Three Resolution Levels. (First View: Initial Render of the Low Resolution Data; 50% and 100%: Percentage (Number of Points) of the Entire Data Set Rendered. The Three Scenes Were Downloaded Simultaneously. Bandwidth is Clamped to 8 Mb/s



Dataset	Cathedral			Patio			Studio			
Num. points	75838	276113	1123222	300405	365155	3000000	314567	315282	442137	670324
First view	464	387	295	620	625	992	356	483	444	294
50%	931	1524	3151	9892	9699	53057	2758	2783	4011	5559
100%	1171	2675	7099	14454	15365	75210	4877	5009	6340	7771

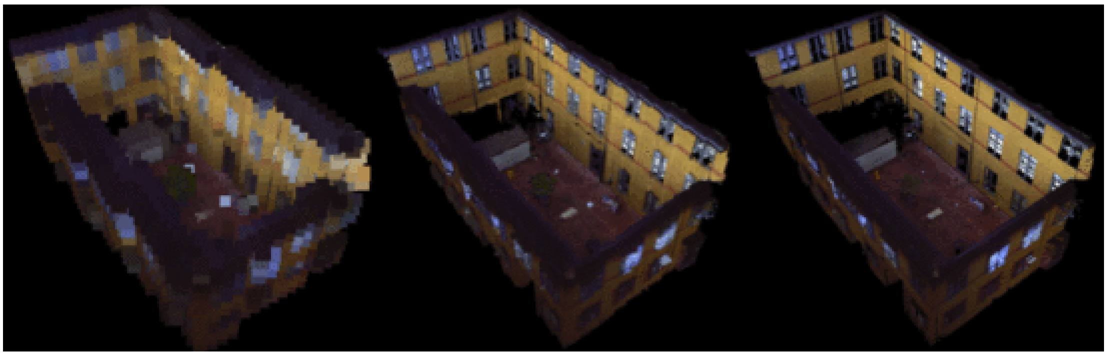


Fig. 11.
Progressive rendering of base LIDAR scan used in Patio scene.

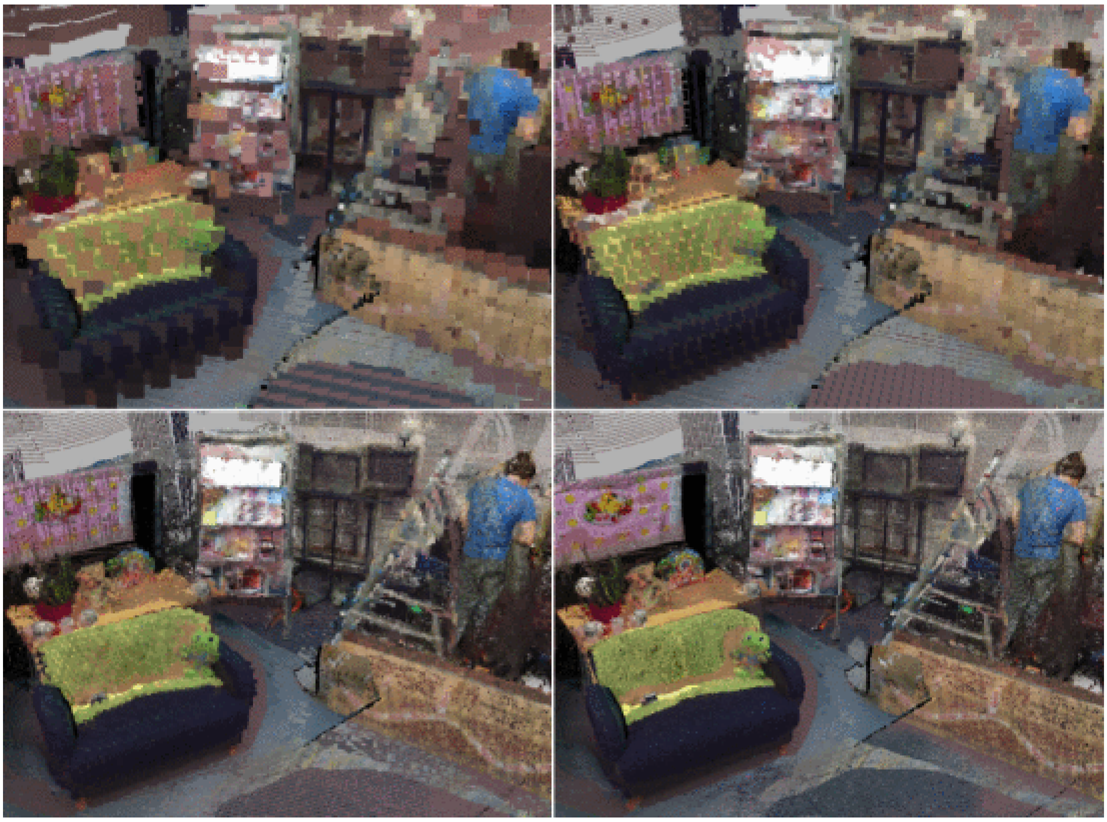


Fig. 12.

Progressive and simultaneous rendering of four point clouds, with resolution increasing from top-left to bottom-right.

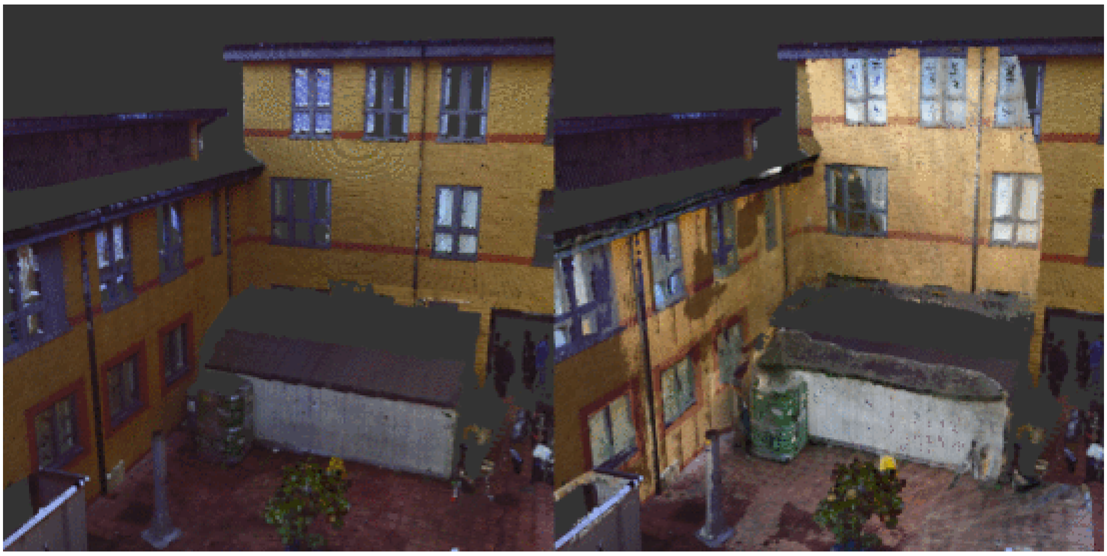


Fig. 13.
Rendering of LIDAR only (left) and LIDAR + still photograph (right).

An interesting comparison of our progressive point cloud rendering method is with that provided by Potree [31]. While implementational details and timings of the Potree method are yet to be published, it clearly uses a similar LOD approach to ours. However, beyond that basic similarity, the techniques appear different. Potree seems designed to minimize the data downloaded by increasing the LOD of those areas which are currently within a certain distance of the camera. While our work does support this feature (see [30]), we choose to disable it for this application, in the interest of downloading the entire data set as quickly as possible—this also makes it unfeasible to compare download times, as Potree makes a point of not downloading the entire data set if possible. We do note, however, that in one of our trial data sets (the largest point cloud from the Patio set), the Potree rendering presents some artifacts between the cells of the hierarchical data structure, which are not present in our work (see Fig. 18).

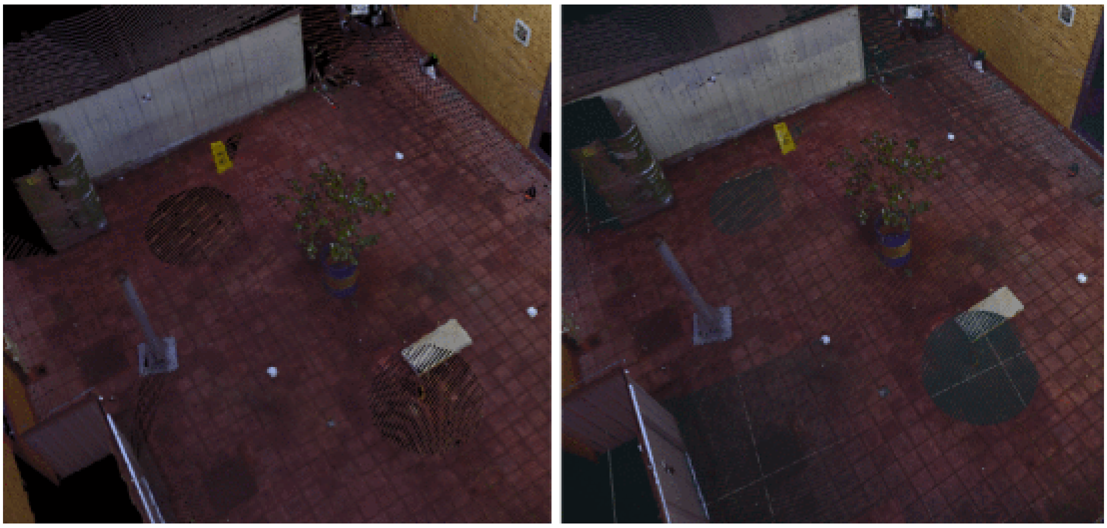


Fig. 18.
Rendering of our method (left) and [31] (right). The latter features white artifacts between the cells used for the data structure.

SECTION IX.

Conclusion

Typically, processing and visualization of big multimodal data are split between individual tools, with video and images processed in a 2D domain then visualized using a thumbnail browsing interface, and 3D data in dedicated 3D production and rendering software. In this paper, we have introduced a framework for unified 3D Web-based visualization of multimodal digital media production data sets, which allows various input modalities to be registered into a unified 3D space, and visualized in hybrid-mode Web application.

A multidomain feature description extended from an existing feature descriptor and a hybrid RANSAC-based registration technique were proposed. The approach was tested on our multimodal database acquired from various modalities including active and passive sensors as well as public single-modal data set. The proposed method shows two times higher precision of feature matching and more stable registration performance than conventional 3D feature descriptors.

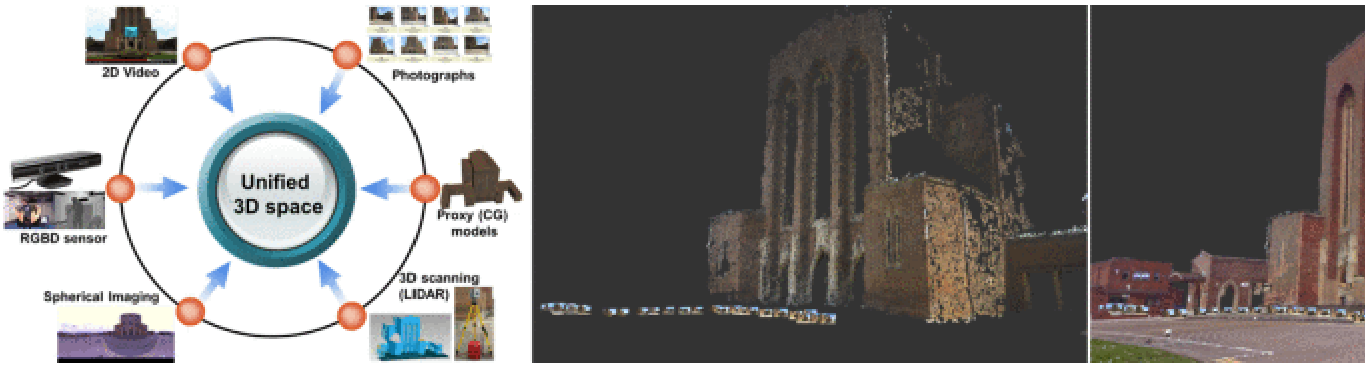
Visualization of production data via the Web is currently become increasingly relevant as modern workflows become based in the cloud. Our Web-based visualization takes advantage of the power of the Web-context to integrate several viewing modalities into a single application, with the additional advantages of the Web: machine independence, no specialized software requirements, viewing from anywhere in the world, etc. The results show that our progressive download method reduces the problems relating to remote viewing of big data. The principal contribution of this aspect of the work is that few other researchers have presented results on progressive visualization of point cloud data via the Web; and (to our knowledge) our work represents the first effort to do so as part of a wider hybrid visualization of multimodal data.

Future work on the multimodal data registration aims to extend to a large-scale spatiotemporal scene data producing a coherent view of the world. It deals with synchronization and registration of multimodal data streams captured by very large and diverse collections of professional and consumer devices under uncontrolled and unpredictable environments. Another direction of extension will be registration of nonvisual data such as audio and text (annotation and metadata). New feature description and matching method for cross-modalities should be developed. Although our current system works effectively on tablet devices, our future work on visualization is now focused on integrating more elements of mixed reality into the application. This possibility is opened due to the fact that mobile versions of many Web browsers allow JavaScript access to the device accelerometer and camera, raising the prospect of remote users being able to visualize a current data set in real-time (i.e., on the same day as the capture) and use of tablet devices as a virtual ‘window’ into the scene, moving it around in space to view the reconstructed scene.

Authors
Figures

Fig. 1.

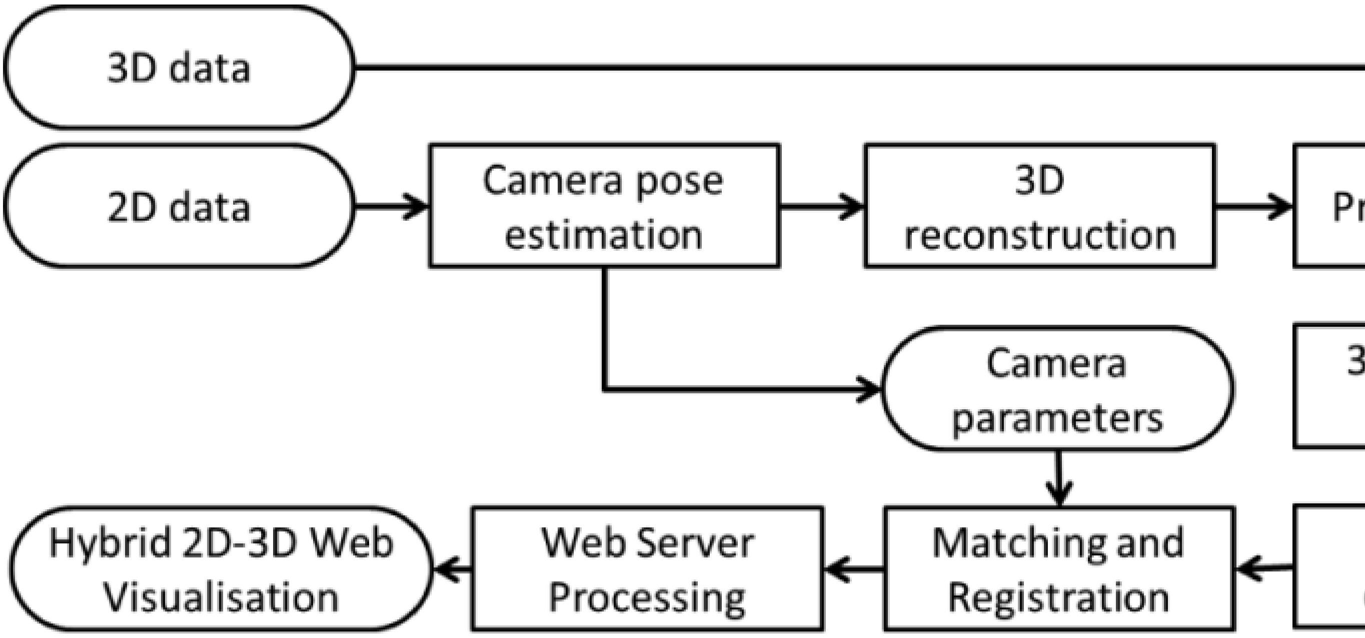
Show in Context



Multimodal data registration and visualization. Left: overview of multimodal visual data registration. Middle: multiple photographs and their 3D reconstruction. Right: registration to LIDAR coordinate system.

Fig. 2.

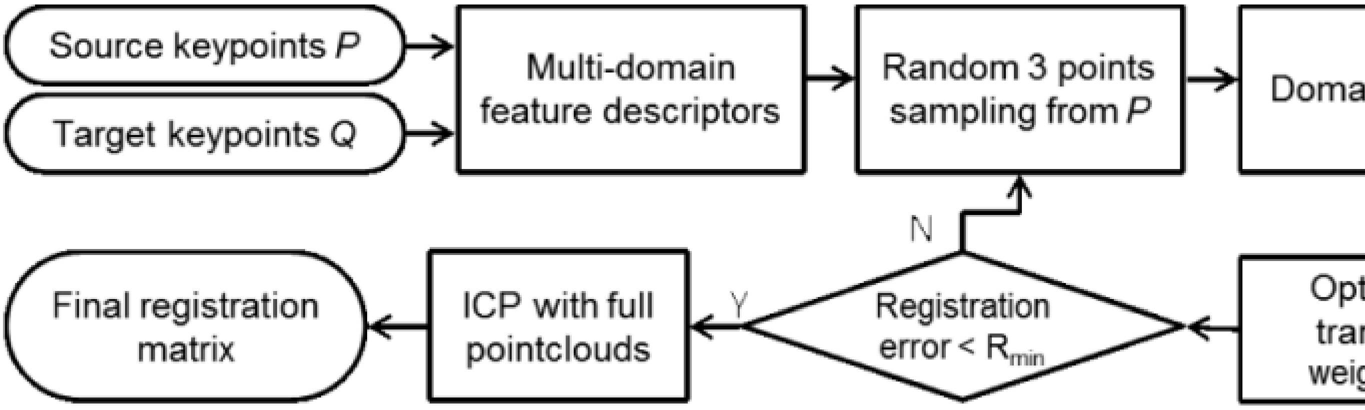
Show in Context



Pipeline for multimodal data registration and visualization.

Fig. 3.

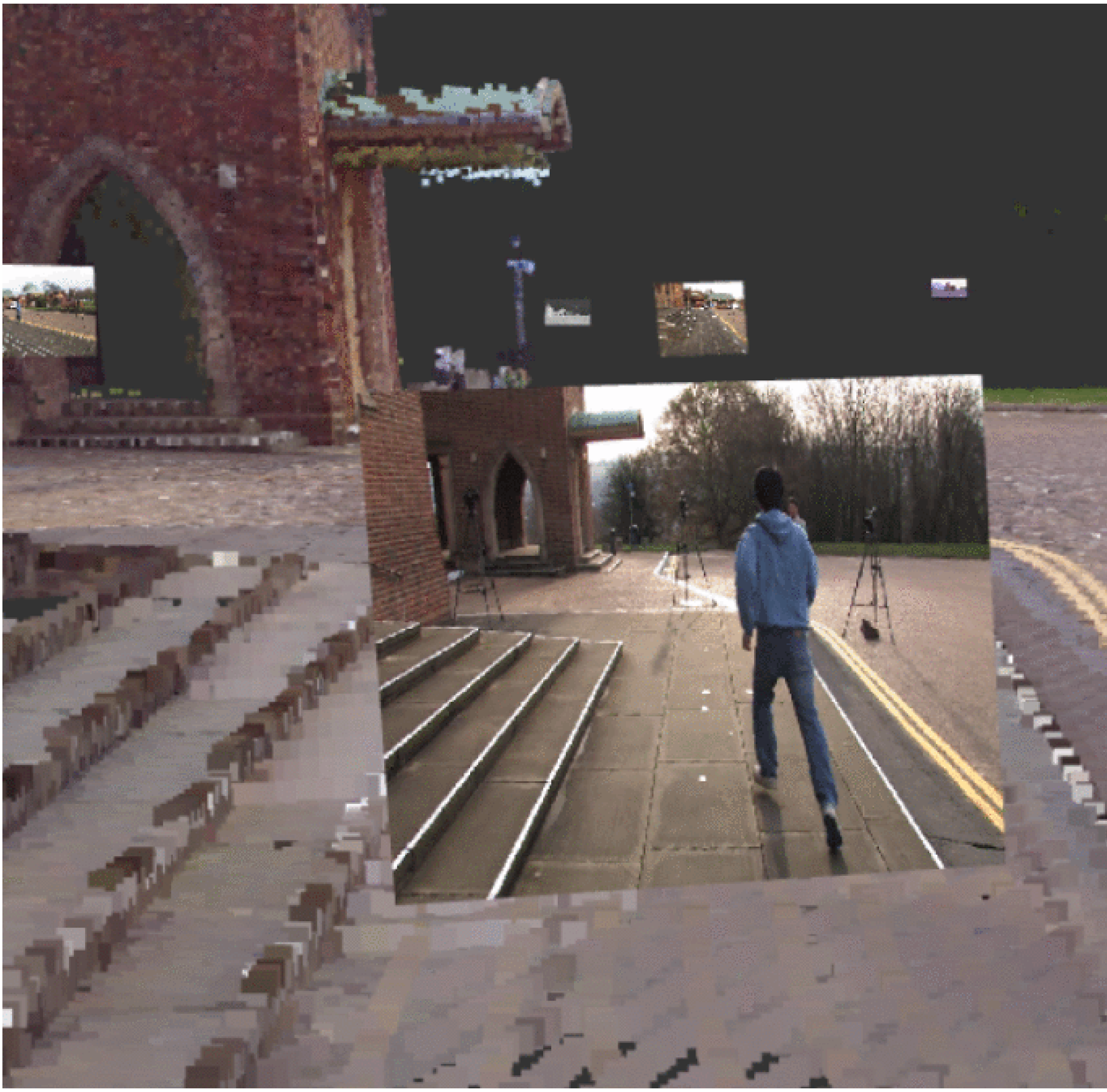
Show in Context


Source keypoints *P*
Target keypoints *Q*
Multi-domain feature descriptors
Random 3 points sampling from *P*
Doma
N
Final registration matrix
ICP with full pointclouds
Y
Registration error < R_{min}
Opt
trar
weig

Hybrid RANSAC-based feature matching and registration.

Fig. 14.

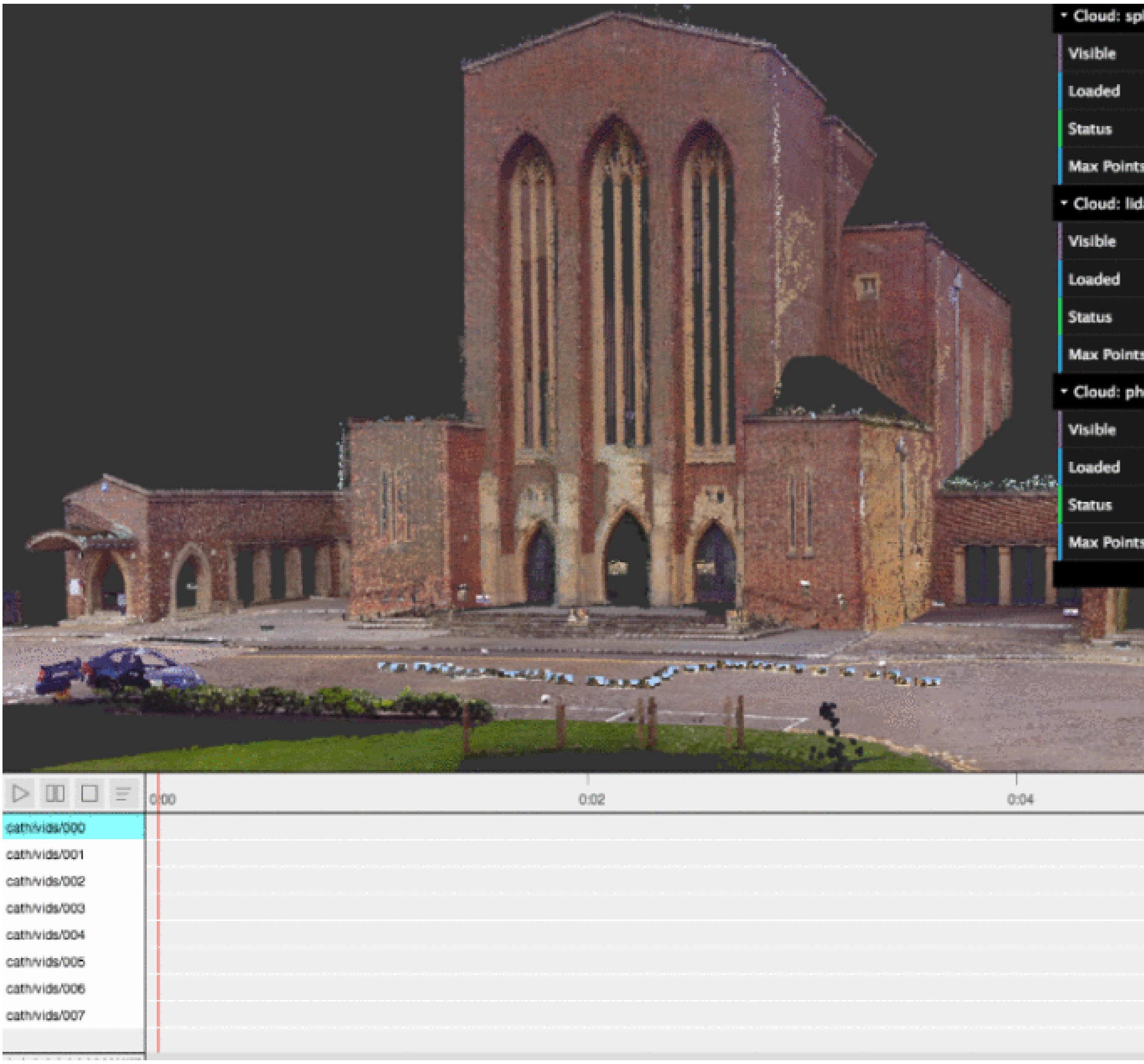
Show in Context



(Hidden) HTML5 video element pipes texture information, at 30 frames/s, positioned to the original camera location and orientation.

Fig. 15.

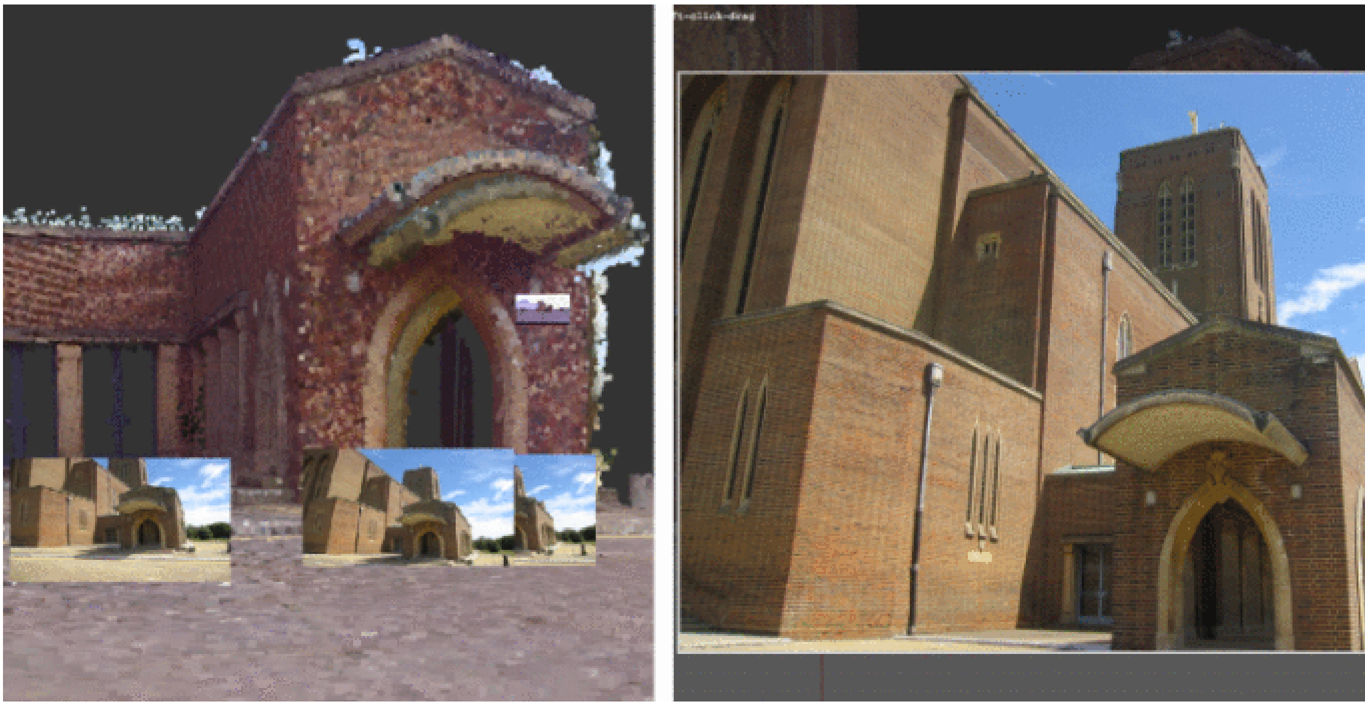
Show in Context



Hybrid 2D–3D Web interface showing the timeline component and GUI overlaying the 3D context.

Fig. 16.

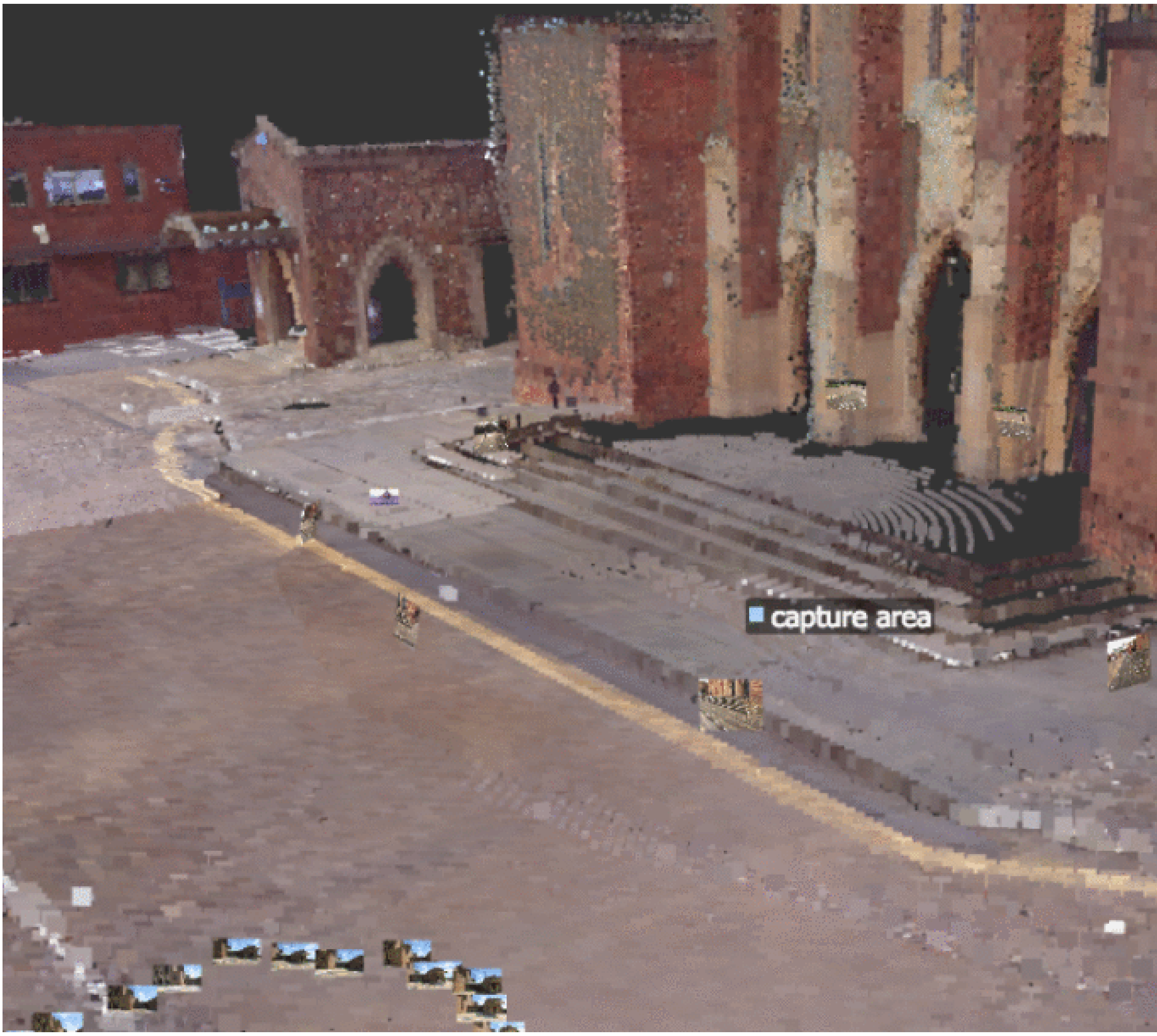
Show in Context



Left: sensors represented as billboards with a thumbnail of original image. Right: clicking on the billboard displays the full resolution image.

Fig. 17.

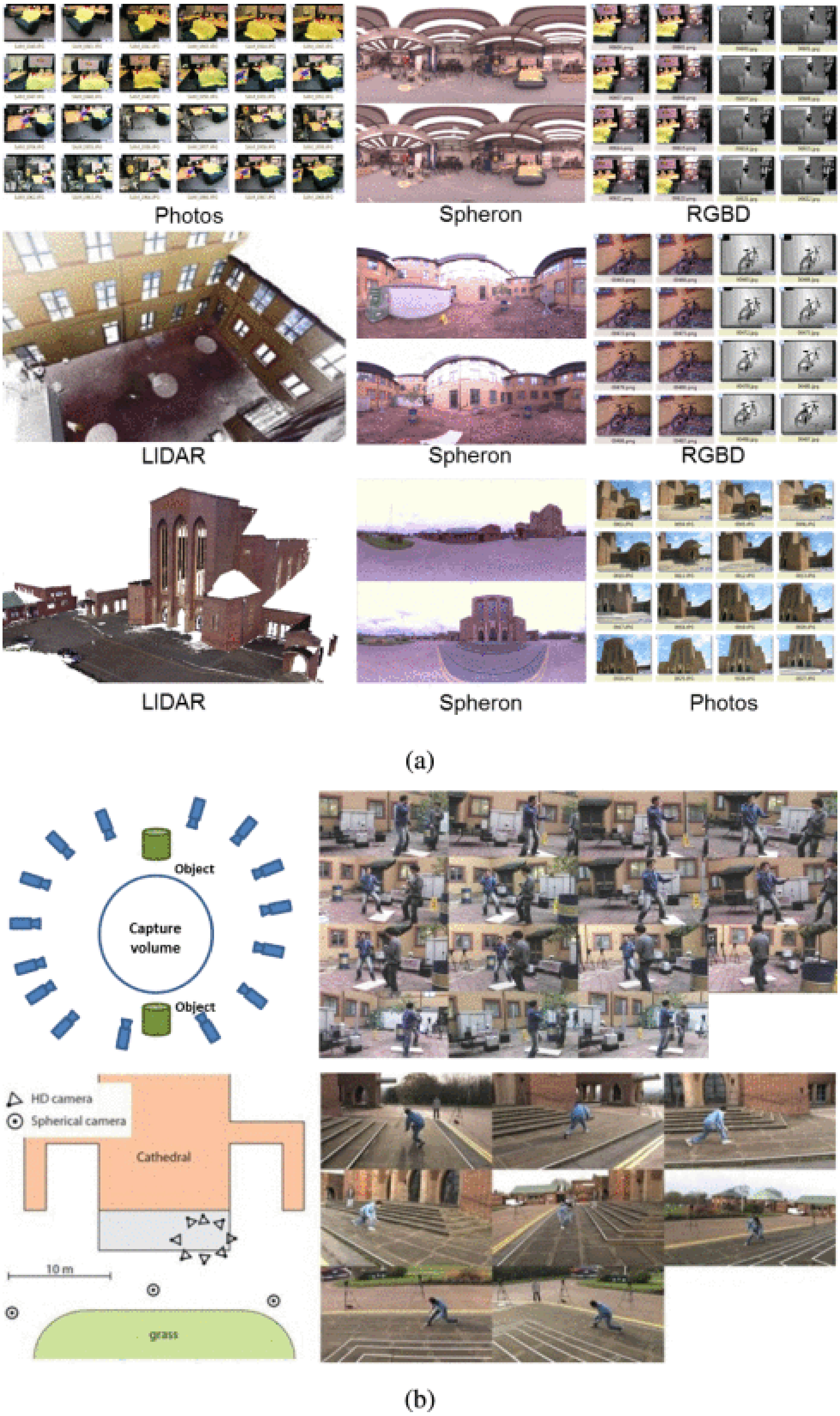
Show in Context



Screenshot showing an example of the annotation component being used to label elements in the scene.

Fig. 4.

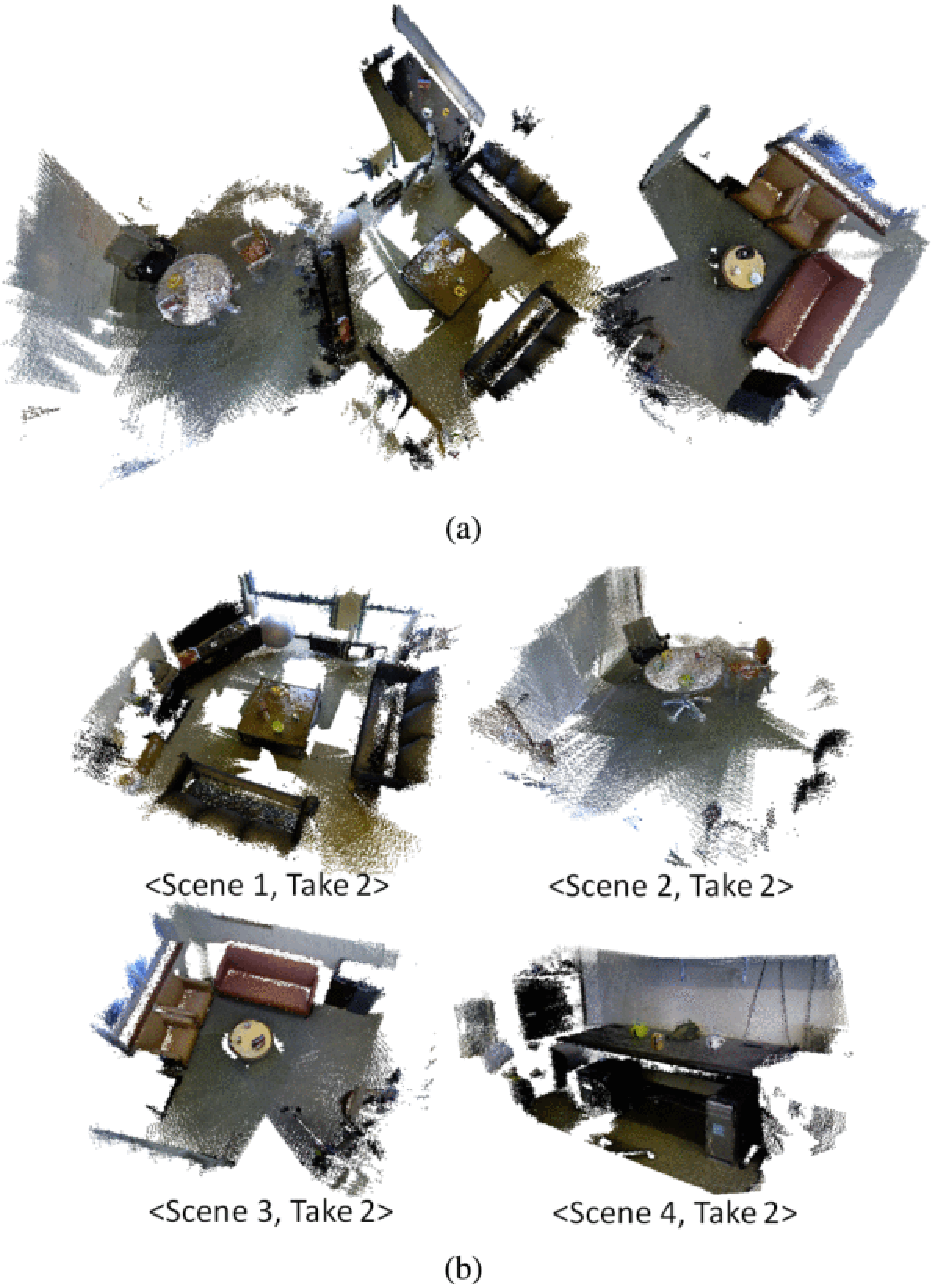
Show in Context



Examples of multimodal data sets. (a) Static scene capture (top: Studio, middle: Patio, bottom: Cathedral). (b) Dynamic scene capture (top: Patio, bottom: Cathedral).

Fig. 5.

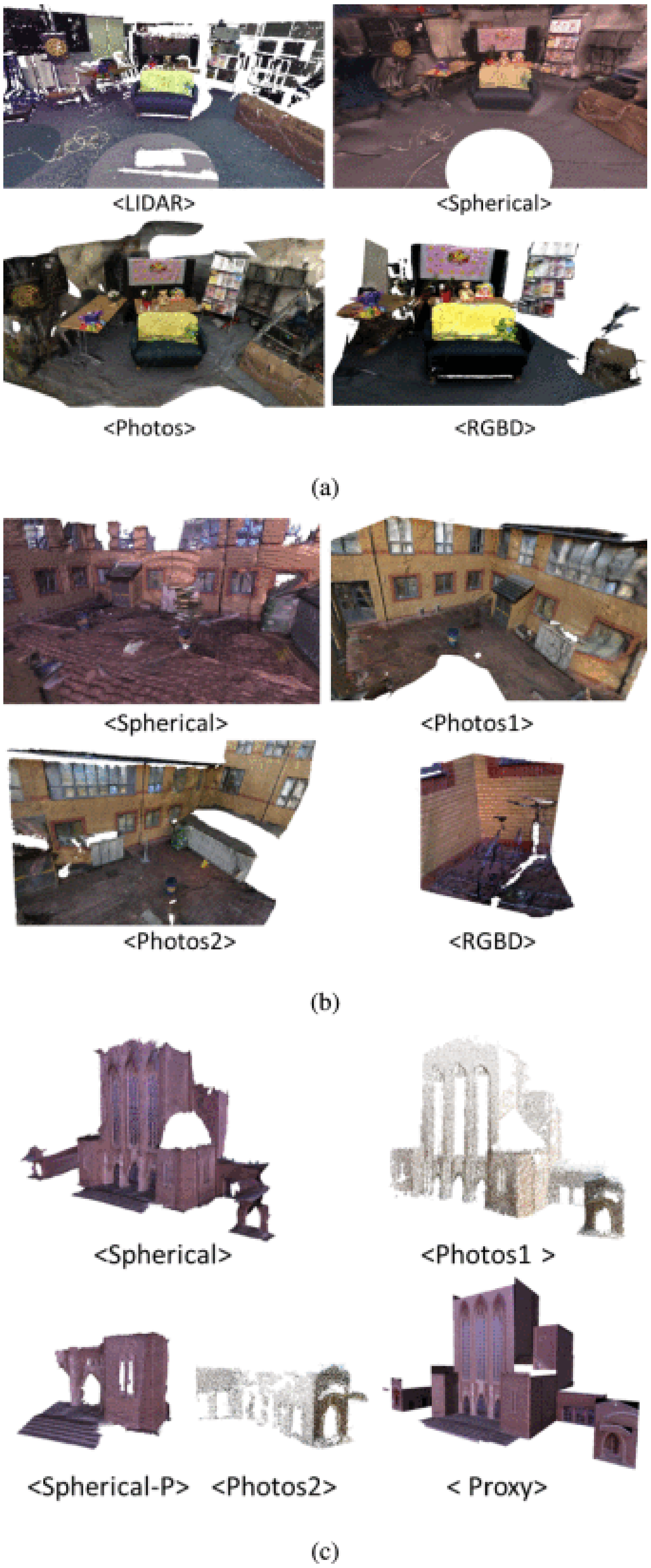
Show in Context



Washington RGB-D scenes data set. (a) Target reference from takes 1. (b) Test sets to be registered.

Fig. 6.

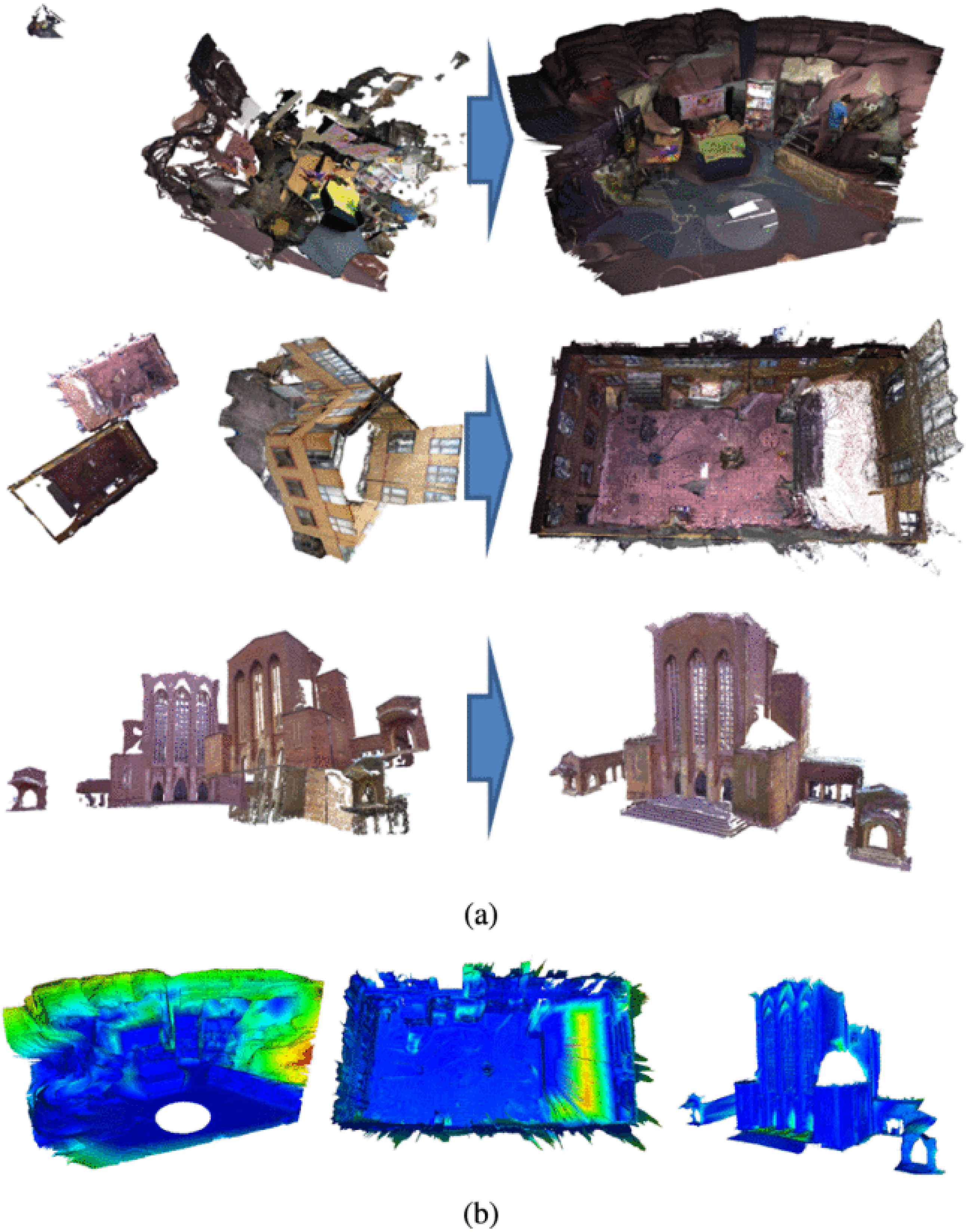
Show in Context



3D models for registration. (a) Studio. (b) Patio. (c) Cathedral.

Fig. 7.

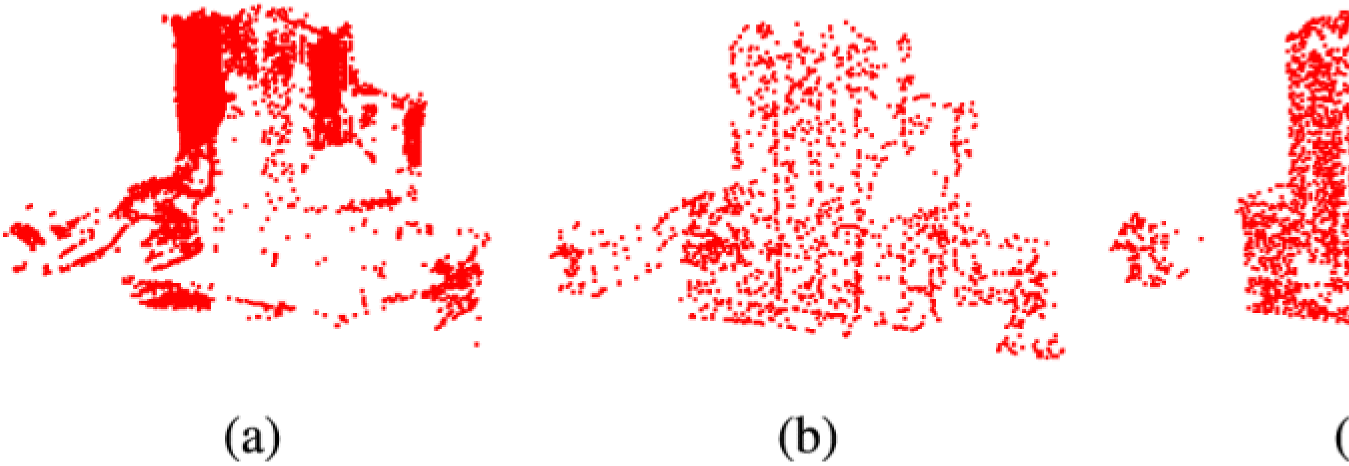
Show in Context



Ground-truth registration. (a) Registration (top: Studio, middle: Patio, bottom: Cathedral). (b) Error map of spherical model (left: Studio, middle: Patio, right: Cathedral).

Fig. 8.

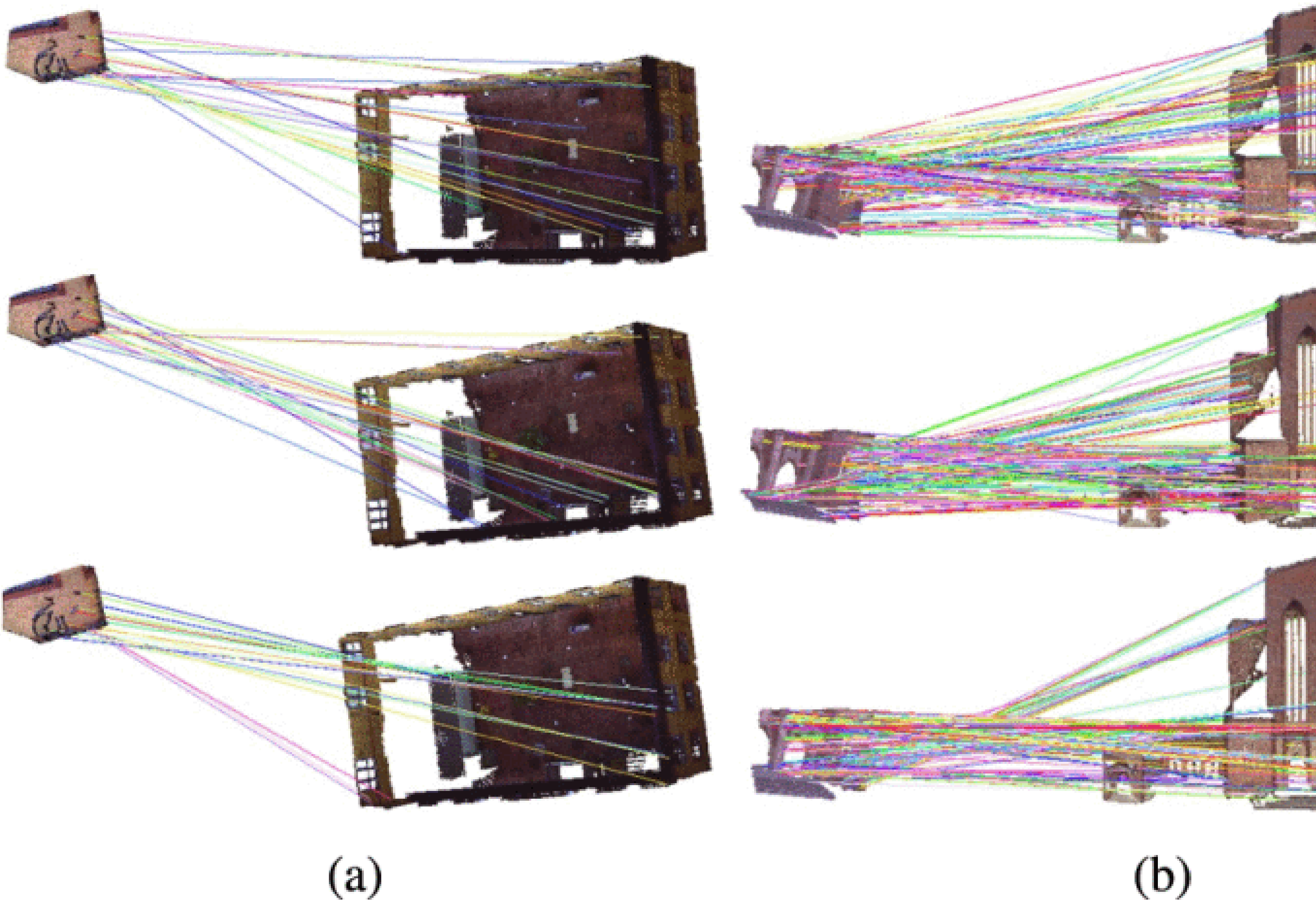
Show in Context



Feature detection result (Cath-S). (a) Noble (9729 points). (b) SIFT (2115 points). (c) Tomasi (2461 points).

Fig. 9.

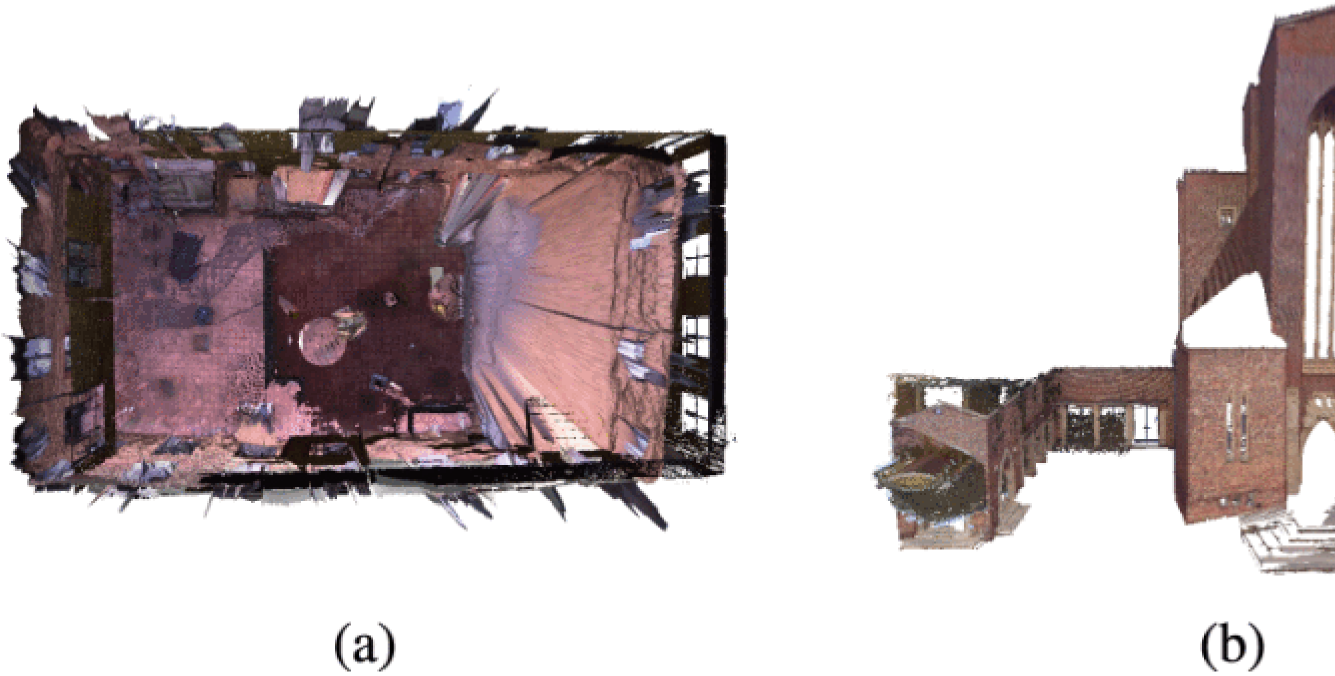
Show in Context



Matched features (top: SHOT, middle: FPFH, bottom: Proposed). (a) Patio-R to LIDAR. (b) Cath-SP to LIDAR.

Fig. 10.

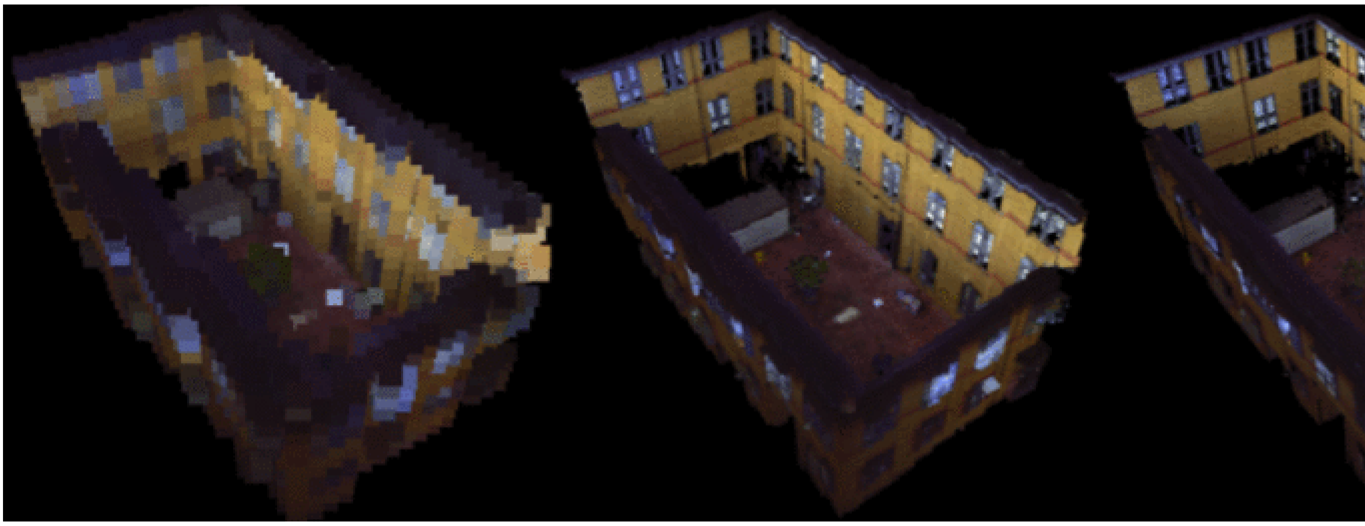
Show in Context



Failure cases in registration. (a) Patio-S with FPFH. (b) Cath-P2 with FPFH$_{LC}$.

Fig. 11.

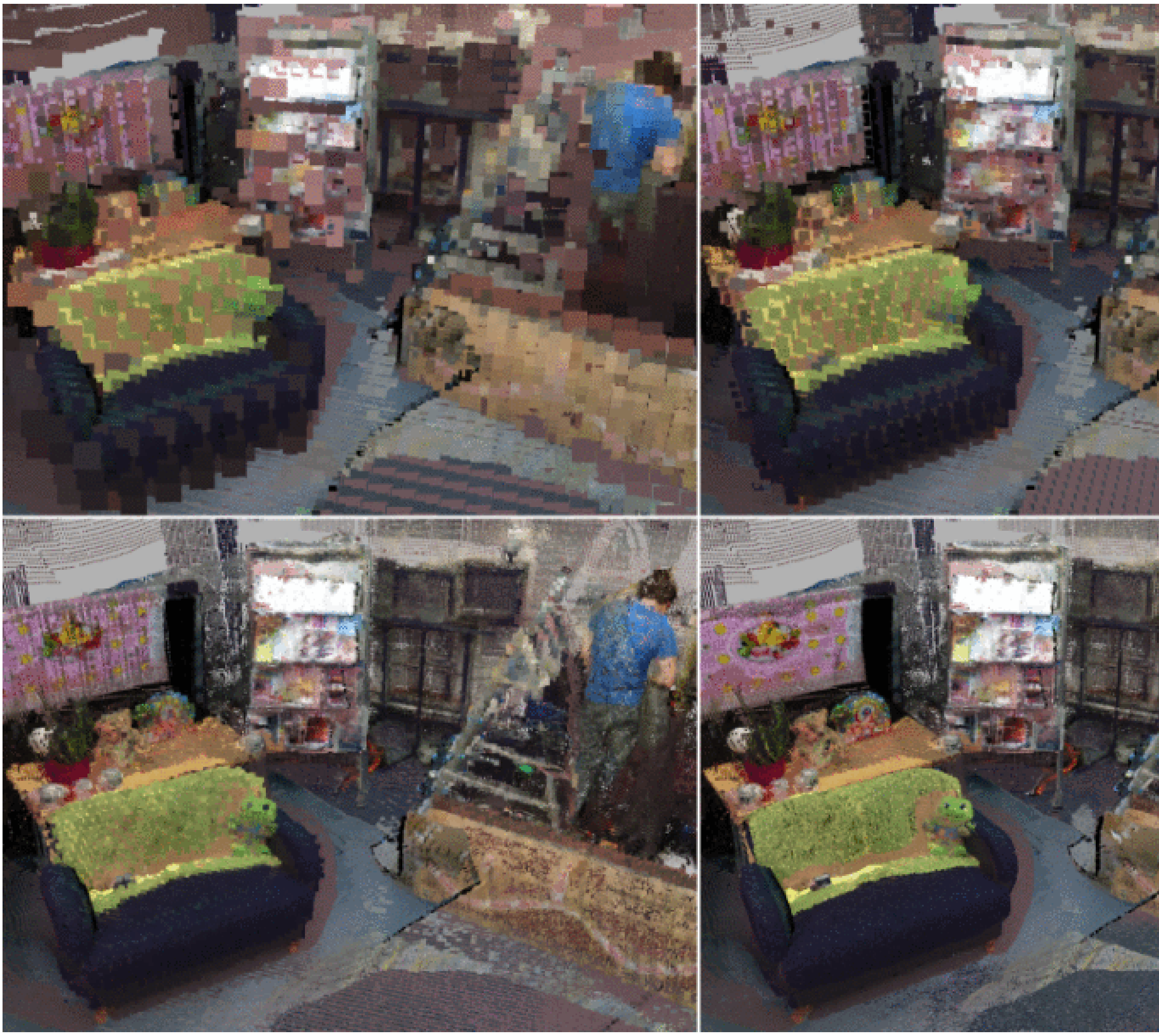
Show in Context



Progressive rendering of base LIDAR scan used in Patio scene.

Fig. 12.

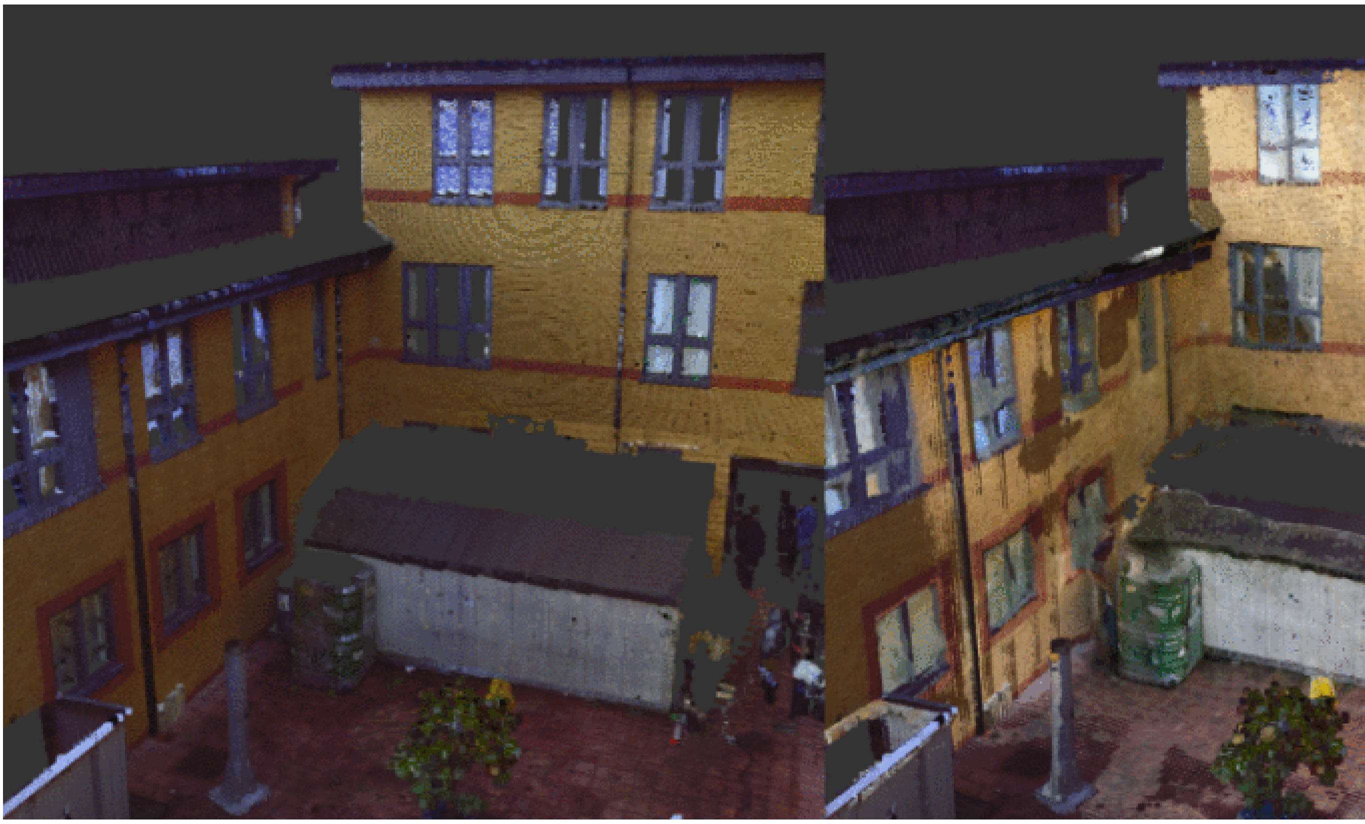
Show in Context



Progressive and simultaneous rendering of four point clouds, with resolution increasing from top-left to bottom-right.

Fig. 13.

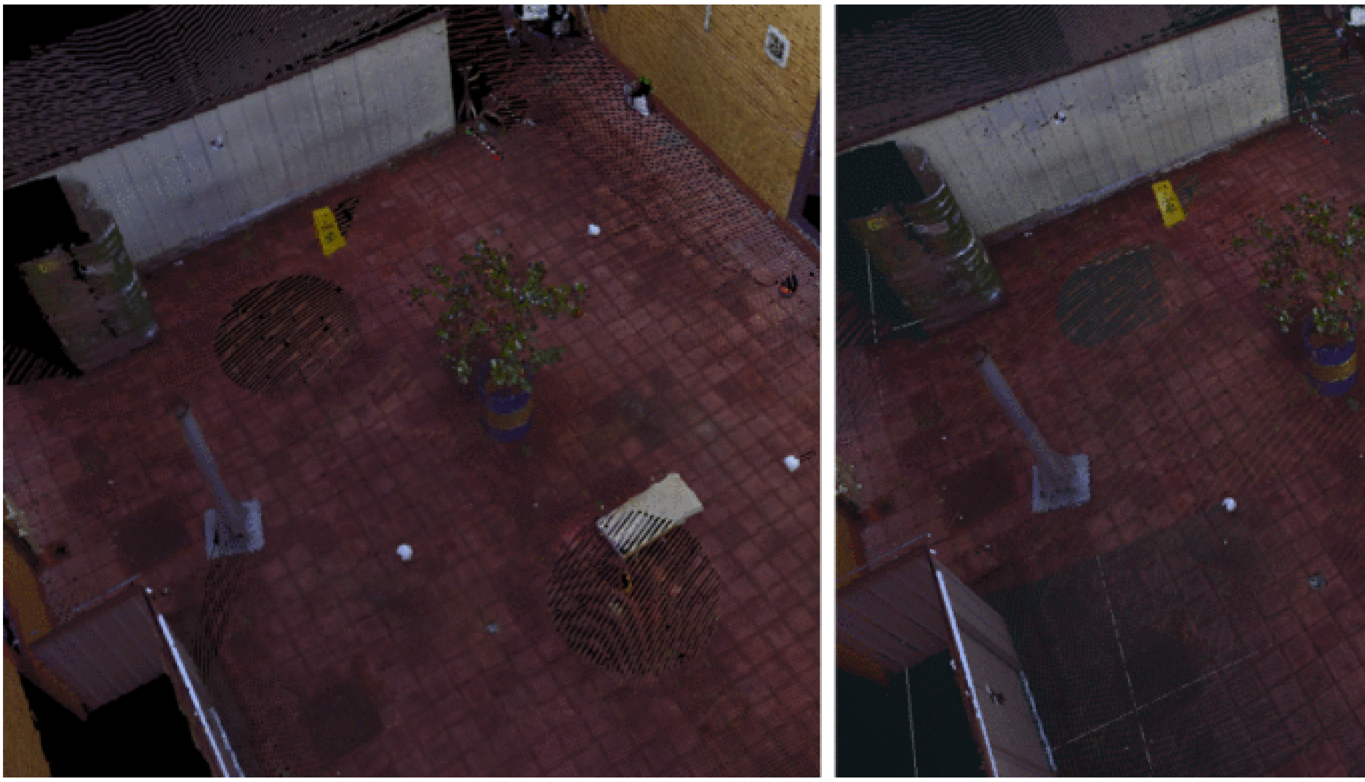
Show in Context



Rendering of LIDAR only (left) and LIDAR + still photograph (right).

Fig. 18.

Show in Context



Rendering of our method (left) and [31] (right). The latter features white artifacts between the cells used for the data structure.