3D Exploitation of 2D Imagery

Peter Cho and Noah Snavely

Recent advances in computer vision have enabled the automatic recovery of camera and scene geometry from large collections of photographs and videos. Such three-dimensional imagery reconstructions may be georegistered with maps based upon ladar, geographic information system, and/or GPS data. Once 3D frameworks for analyzing two-dimensional digital pictures are established, high-level knowledge readily propagates among data products collected at different times, places, and perspectives.

We demonstrate geometry-based exploitation for several imagery applications of importance to the defense and intelligence communities: perimeter surveillance via a single stationary camera, rural reconnaissance via a mobile aerial camera, urban mapping via several semicooperative ground cameras, and social media mining via many uncooperative cameras. Though a priori camera uncertainty grows in this series and requires progressively more computational power to resolve, a geometrical framework renders all these applications tractable.

» **The invention of the digital camera is** attributed to Eastman Kodak engineer Steven Sasson in 1975 [1]. Sasson's device recorded 0.01-megapixel black-and-white pictures to cassette tapes. The first digital photograph required 23 seconds to generate and needed a television set to display [2]. Today, digital cameras recording greater than 10-megapixel color photos to Secure Digital (SD) cards with multi-gigabyte storage capacities are commonplace. Though the total number of electronic images in existence is not known, billions of photos and video clips are now accessible on the World Wide Web.

The current capability to collect and store digital images vastly outpaces the current capability to mine digital pictures. Existing archives of photos and videos are basically unstructured. As anyone who has ever tried to find some particular view of interest on the Internet knows, querying imagery websites can be a frustrating experience. Text-based searches generally do not return salient metadata, such as camera geolocation, scene identity, or target characteristics. Some basic organizing principle is consequently needed to enable efficient navigating and mining of vast digital imagery repositories.

Fortunately, three-dimensional (3D) geometry provides such an organizing principle for imagery collected at different times, places, and perspectives. For example, a set of photos of some ground target represents two-dimensional (2D) projections of 3D world space onto a variety of image planes. If the target's geometry is captured in a 3D map, it can be used to mathematically relate different ground photos of the target to each other. Moreover, as the diagram in Figure 1 indicates, the 3D map connects

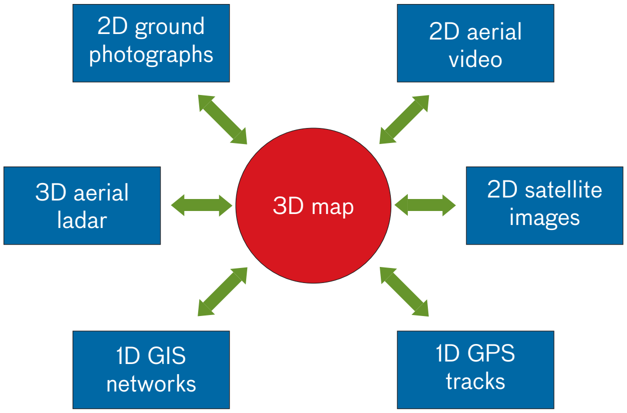


FIGURE 1. Three-dimensional maps provide geometrical frameworks for organizing intelligence collected at different times, places, and perspectives.

together information collected by completely different sensors. Therefore, a photo of the target shot by a ground camera can be related to a corresponding aerial view or ladar image, provided all these data products are georegistered with the 3D map. The map itself acts as a repository of high-level intelligence distilled from multiple sensors. Ultimately, situational awareness comes much more directly from knowledge stored within the map than from the millions of low-level pixels and voxels on which it is based.

In this article, we present a 3D approach to exploiting 2D imagery that follows the flow diagram in Figure 2. Working with photos and video clips originating from diverse sources, we first reconstruct their cameras' relative positions and orientations via computer vision techniques. Such methods geometrically organize a priori unstructured sets of input images. We next georegister reconstructed digital pictures with laser radar (ladar), geographic information system (GIS) layers, and/or Global Positioning System (GPS) tracks to deduce their absolute geocoordinates and geo-orientations, which cannot be determined from pixel contents alone. In all cases we have investigated, good alignment has been achieved between independent datasets often collected years apart and by fundamentally different sensing modalities. Once a 3D framework for analyzing 2D imagery is established, many challenging exploitation problems become mathematically tractable. In this article, we focus upon automatic scene annotation, terrain mapping, video stabilization, target geolocation, urban modeling, indoor/outdoor view connection, photo segmentation, picture geoquerying, and imagery retrieval.

Three-dimensional imagery exploitation can be applied to a wide range of problems of significant importance to the defense and intelligence communities. Figure 3 categorizes such applications according to their a priori camera uncertainty on the horizontal axis and processing complexity on the vertical axis. The applications begin in the figure's lower left corner with perimeter monitoring by a single stationary camera. By enabling narrow fields of view to be mosaiced together into a panoramic backdrop, geometry yields synoptic context for "soda-straw" security camera footage. Data processing becomes more complex for video shot from a mobile surveillance platform such as an unmanned aerial vehicle (UAV). For such reconnaissance imagery, 3D geometry relates views gathered by the moving camera at different places and perspectives. Exploitation grows substantially more complicated for imagery shot by several cameras whose geolocations are not initially known. Finally, data processing requirements turn formidable for Internet pictures collected by uncooperative cameras as their numbers exponentially increase. Nevertheless, substantial geometry information can be recovered even in this most difficult case. Intelligence may then be propagated among images taken by different people at different times with different cameras.

This article considers all the applications depicted in Figure 3, starting from the simplest in the lower left-hand corner and progressing through the hardest. We

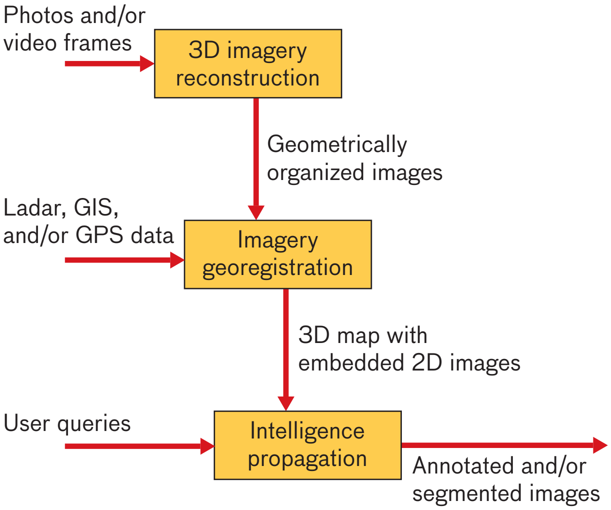

Photos and/or video frames
3D imagery reconstruction
Geometrically organized images
Ladar, GIS, and/or GPS data
Imagery georegistration
3D map with embedded 2D images
User queries
Intelligence propagation
Annotated and/or segmented images

FIGURE 2. Algorithm flow for geometry-based exploitation of digital imagery.

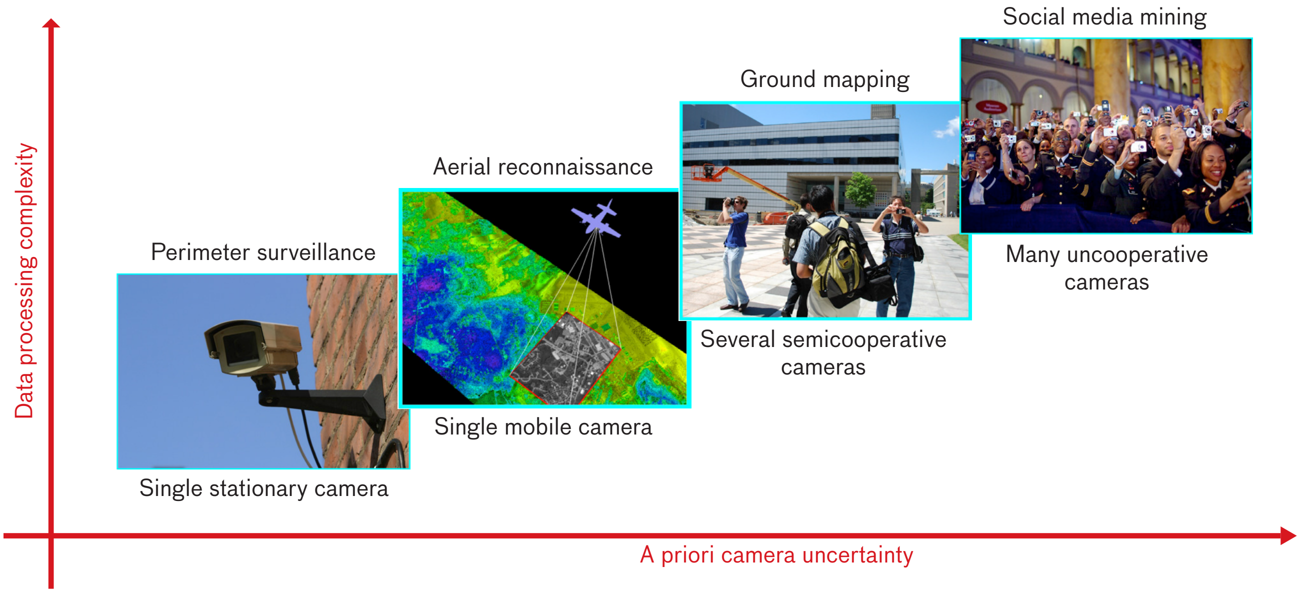


FIGURE 3. Applications of 3D imagery exploitation schematically categorized according to their a priori camera uncertainty and processing complexity. This article presents research on these four applications, progressing from simple perimeter surveillance to the formidably difficult problem of social media mining.

begin with an automated procedure for constructing 3D mosaics from images collected by a stationary camera. Dynamic video streams are matched with static mosaics to provide human observers with useful context. We next move to applications that make use of a mobile camera and exploit imagery gathered by a UAV to generate 3D terrain maps and mark geopoints of interest. Generalizing our techniques to images collected by multiple cameras, we reconstruct urban scene geometry and camera parameters for thousands of ground photos shot semicooperatively around the Massachusetts Institute of Technology (MIT) campus. After georegistering the reconstructed photos to an abstracted city map, we refine 3D building models by texturing orthorectified mosaics onto their facades. Finally, we work with photos of New York City (NYC) downloaded from the web. Once the Internet images are reconstructed and geoaligned to a detailed 3D map, we demonstrate automated labeling of buildings, target point mensuration, image region classification, and ranking of NYC pictures based upon text search queries.

Perimeter Surveillance via a Single Stationary Camera

We begin our imagery exploitation survey with problems including a single camera whose position is constant and known. Although such sensing setups are highly constrained, 3D geometry nevertheless yields useful and surprising imagery results.

3D Mosaicing of Photos and Video Frames

Security video monitoring has grown commonplace as camera technology has increased in quality and decreased in price. Cameras fixed atop poles or attached to building sides are routinely used to follow movements within their fields of view. Such sensors can rotate about their attachment points and zoom to provide close-up views. However, they do not simultaneously yield synoptic context that would help humans better understand their instantaneous output. It would be useful to embed a security camera's dynamic imagery inside a panoramic mosaic covering its accessible field of regard. Therefore, we developed such a visualization capability working with imagery gathered in 2008 from atop the 21-story Green Building on MIT's campus (Figure 4).

Two example photos shot from a tripod-mounted, rooftop camera are presented in Figure 5. Extracting features from such overlapping stills on the basis of their intensity contents represents the first step in generating mosaics [3, 4]. Over the past decade, the Scale Invariant Feature Transform (SIFT) has become a standard method for detecting and labeling features. SIFT is relatively insensitive to varying camera perspectives, zoom levels, and illu-

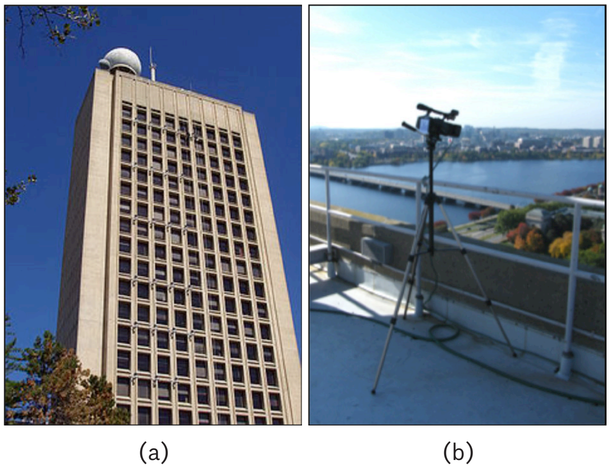


FIGURE 4. Setup for initial experiments that mimic security camera monitoring. (a) Imagery was collected from atop the Green Building on MIT's campus. (b) Stills and video frames were shot from cameras set upon a fixed, rooftop tripod.

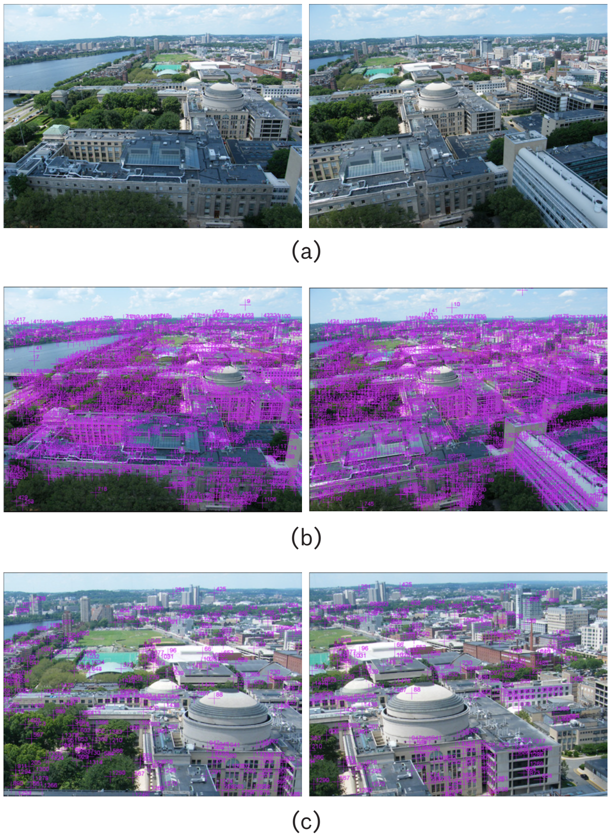


FIGURE 5. (a) Two photos shot from the Green Building's rooftop. (b) 15,428 and 16,483 SIFT features were extracted from the two photos. Only 10% of all SIFT features are displayed. (c) 3523 tiepoints matched between the two overlapping photos. Only 10% of all tiepoint pairs are displayed in these zoomed views.

mination conditions [5]. It also yields a 128-dimensional vector descriptor for each local feature. Figure 5b illustrates SIFT output for the two rooftop photos in Figure 5a.

Looping over pairs of input photos, we next identify candidate feature matches via Lowe's ratio test [5]. Using approximate nearest-neighbor data structures to significantly decrease search times over the 128-dimensional vector space [6, 7], our machine computes distances d_1 and d_2 between the closest and next-to-closest candidate partner descriptors in a photo pair. We accept the closest feature as a genuine tiepoint candidate if $d_1/d_2 < 0.5$.

A number of incorrect feature matches slip through Lowe's ratio filter. Thus, an iterative random sample consensus (RANSAC) algorithm is employed to identify erroneous pairings [8]. Four sets of tiepoints randomly pulled from different image quadrants are used to construct homography transformations that relate image plane pairs. All surviving features in one photo are projected via each randomly generated homography onto the second photo. If the distance between a projected feature and its candidate tiepoint is sufficiently small, the features are counted as an inlier pair. The homography maximizing the inlier count serves as the defining classifier of SIFT outliers. Final pairs of inliers are relabeled so that they share common identifications. Figure 5c illustrates feature-matching results for the two photos in Figure 5a.

After features have been matched across all input photographs, our machine moves on to sequentially form mosaics from subsets of images ordered by decreasing tiepoint pair count. We assume every photo's intrinsic camera calibration parameters are known except for a single focal length [9, 10]. The linear size of the camera's charge-coupled-device (CCD) chip, along with its output metadata tags, provides initial estimates for each photo's dimensionless focal parameter. Three-dimensional rays corresponding to 2D tiepoints are calculated, and a matrix is formed by summing outer-products of associated rays. Singular value decomposition of this matrix yields a rough guess for the relative rotation between image pairs [4, 11].

Armed with initial estimates for all camera calibration parameters, we next perform iterative bundle adjustment using the LEVMAR package for nonlinear Levenberg-Marquardt optimization [12]. When refining focal and rotation parameter values for the nth image, our machine holds fixed the fitted camera parameters for the previous $n - 1$

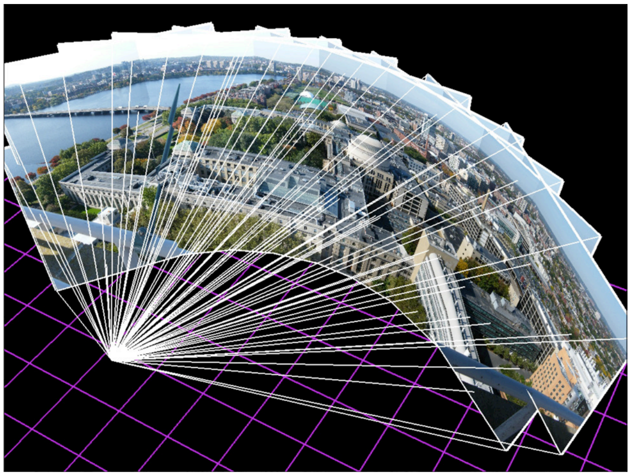


FIGURE 6. A 3D mosaic of 21 tripod photos shot from the rooftop of MIT's Green Building. [video attached, check here]

images. After all photos have been added to the composite, we perform one final bundle adjustment in which all camera focal and rotation parameters are allowed to vary.

Figure 6 displays results from this mosaicing procedure for 21 photos shot from the MIT skyscraper rooftop. Three-dimensional frusta depict the relative orientation and solid angle for each 2D image in the figure. No attempt has been made to blend colors within the overlapping ensemble of photos. Nevertheless, the entire collection yields a high-fidelity, wide-angle view of a complex scene.

After shooting stills for the panoramic mosaic, we replaced the digital camera on the rooftop tripod with a high-definition video camera. Video footage was then collected inside the panorama's field of regard. To demonstrate a future capability for augmenting security camera output in real time, we want to match each foreground video frame with the background mosaic as quickly as possible. We therefore developed the following algorithm whose performance represents a compromise between accuracy and speed.

For each video frame, we extract SIFT features and match them with counterparts in the mosaiced photos that were precalculated and stored. If a panoramic tiepoint partner is found for some video feature, a 3D ray is generated from the calibrated still and associated with the feature's 2D video coordinates. An iterative RANSAC procedure similar to the one employed for static panorama generation is utilized to minimize false correspondences between ray and coordinate pairs. The homography that maps 3D world-space rays onto 2D video feature coordinates is subsequently determined via least-squares fitting. The entries in the homography are transferred to a projection matrix for the camera, which is assumed to reside at the world-space origin. All intrinsic and extrinsic camera parameters for each video frame may then be recovered from its corresponding projection matrix. This process independently matches each foreground video image to the background panorama.

Figure 7 exhibits the frustum for the video camera embedded among the frusta for the mosaiced stills at one instant in time. In order to emphasize that the former is dynamic while the latter are static, we recolor the pan-

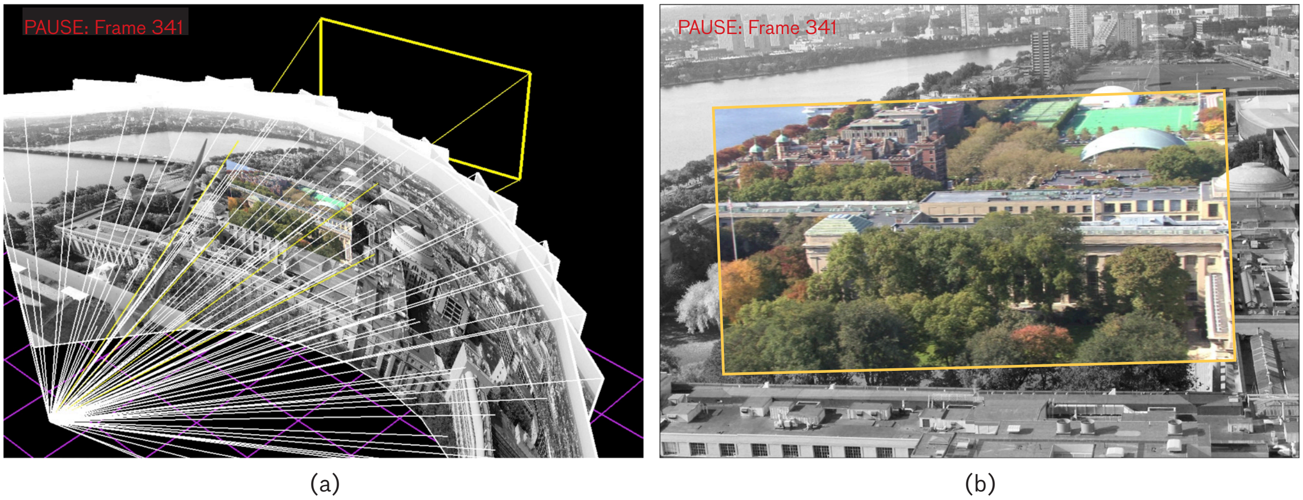
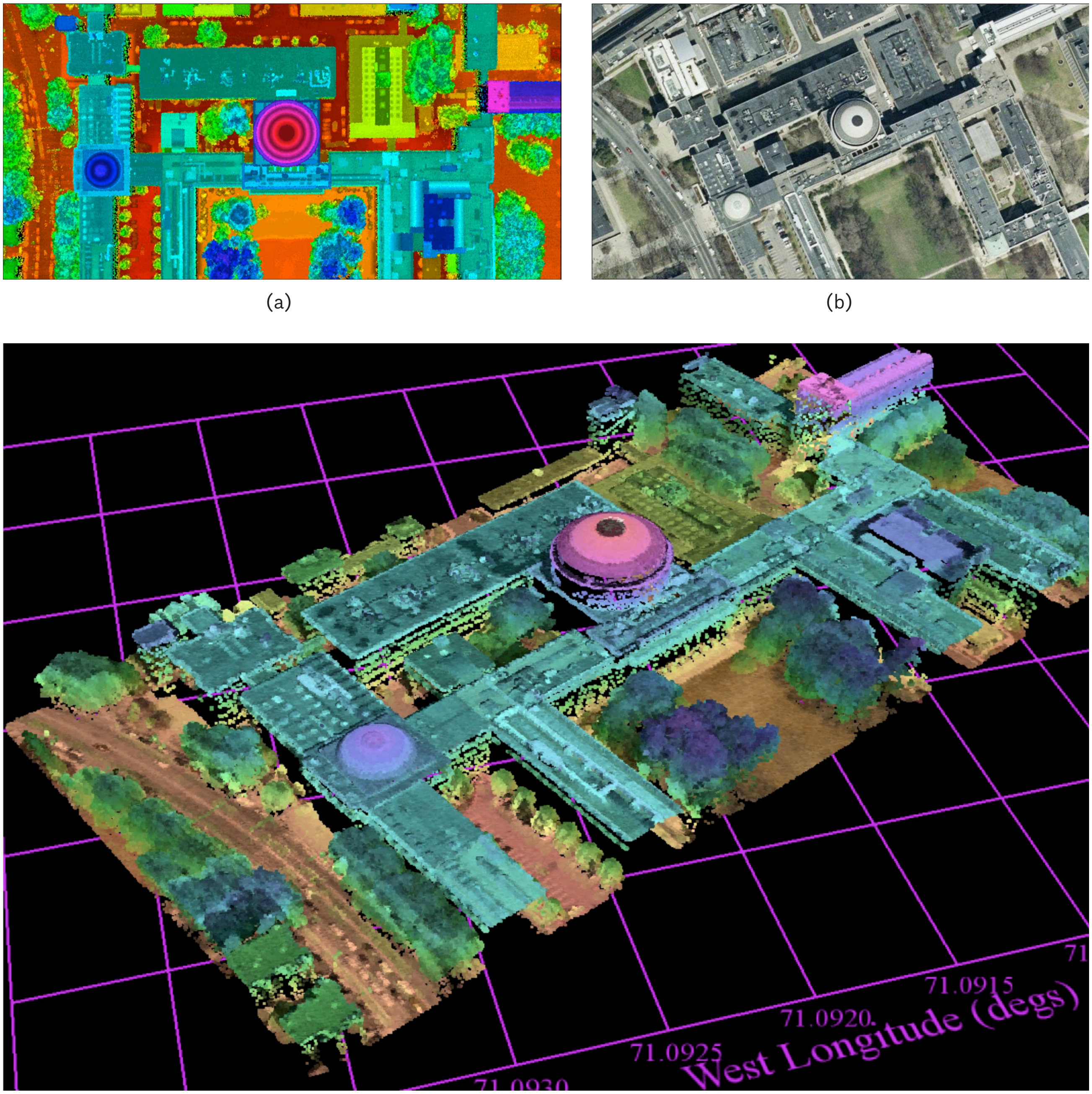


FIGURE 7. Dynamic video versus static mosaic. (a) The instantaneous angular location of one video frame relative to the background mosaic is indicated by its yellow frustum. (b) The colored video frame is aligned with the gray-scale panorama viewed from the rooftop tripod's perspective. [video]


(a)

(b)

(c)

FIGURE 8. (a) Aerial ladar point cloud colored according to height. (b) Aerial photograph naturally colored. (c) Aerial ladar and electro-optical imagery fused together within a 3D map.

orama pictures on a gray scale so that the colored video image stands out. We also temporally smooth the projection matrices for every video frame via an αβγ filter [13, 14]. The video frames then glide over the panorama with minimal jitter and yet keep up with sudden changes in camera pan and tilt. As the movie plays and roams around in angle space, it may be faded away to reveal good agreement between the soda-straw and synoptic views.

The absolute geoposition, geo-orientation, and scaling of the frusta in Figure 7 cannot be determined by conventional computer vision techniques alone. To fix these global parameters, the photos and video frames must be inserted into a world map. We therefore next consider 3D geoalignment of 2D panoramas.

Georegistering Mosaics

The geometry of outdoor environments is generally complex. Fortunately, 3D terrain can be efficiently measured by aerial laser radars. High-resolution ladar imagery is now routinely gathered via airborne platforms operated by government laboratories and commercial companies. Ladars collect hundreds of millions of points whose geolocations are efficiently stored in and retrieved from multiresolution tree data structures. Laser radars consequently yield detailed underlays onto which other sensor measurements can be draped.

Figure 8a illustrates an aerial ladar point cloud for a section of MIT's campus. These data are colored according to height via a color map designed to accentuate Z-content. Figure 8b exhibits a conventional aerial image snapped from Yahoo's website [15]. The snapshot covers the same general area of MIT as the ladar map. The 2D photo captures panchromatic reflectivities, which the 3D ladar image lacks. To maximize information content, we fuse the two together using an HSV (hue, saturation, and value) coloring scheme [16]. The fused result is displayed on a longitude-latitude grid in Figure 8c.

In winter 2009, we shot a second sequence of 14 ground photos from MIT's student union, which is located near the lower left of Figure 8c. Following the same mosaicing procedure as for our first set of images collected from atop the Green Building, we converted the overlapping 2D pictures into 3D frusta (Figure 9). Given the Yahoo aerial photo, it is relatively straightforward to geolocate the cameras within the ladar map. On the other hand, computing the multiplicative factor by which each frustum's focal length needs to be rescaled is more involved. Technical details for the scale factor computation are reported in Cho et al. [17].

In order to align the photo mosaic with the ladar point cloud, we also need to compute the global rotation R_{global}, which transforms the rescaled bundle of image-space rays onto its world-space counterpart. We again form a matrix sum of corresponding ray outer-products and recover R_{global} from its singular value decomposition [4, 11]. After the camera projection matrix for each mosaiced photo is rotated by R_{global}, we can insert the panorama into the 3D map (Figure 10).

Though the absolute geoposition, geo-orientation, and scale of the photos' frusta are fixed, the range at which the image planes are viewed relative to the camera's location remains a free variable. By varying this radial parameter, we can visually inspect the alignment between the ground-level pictures and aerial ladar data. In Figure 11a, the image planes form a ring relatively close to the ground camera and occlude most of its view of the ladar point cloud. When the ring's radius is increased as in Figure 11b, some ladar points emerge in front of the image planes from the ground tripod's perspective. It is amusing to observe green leaves from the summertime ladar data "grow" on nude tree branches in the wintertime photos. The striking agreement between the tree and building contents in the 2D and 3D imagery confirms that the mosaic is well georegistered.

Once the panorama is aligned with the point cloud, ladar voxels match onto corresponding photo pixels.

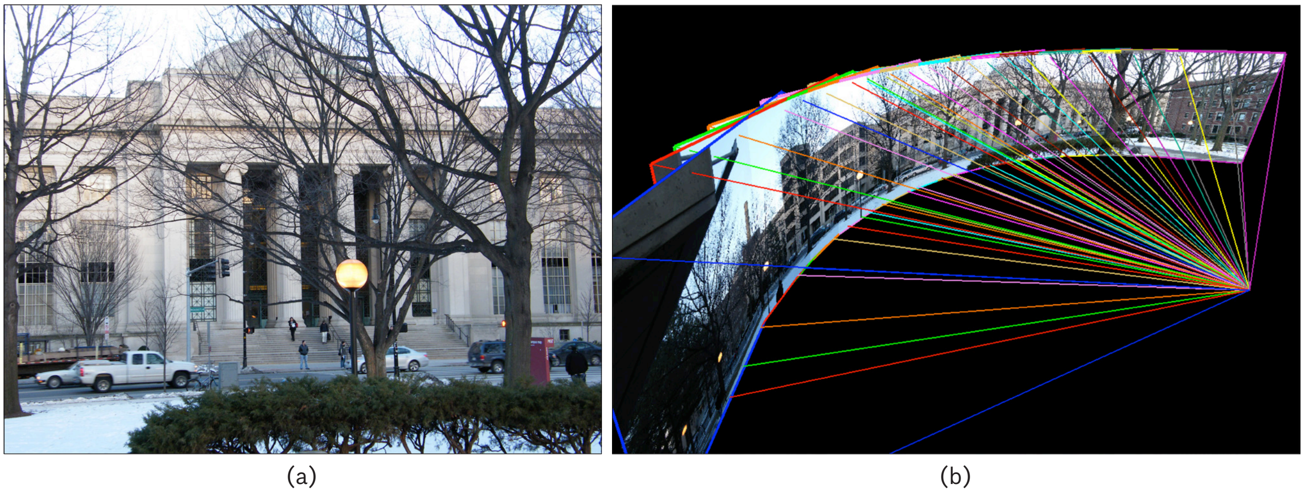


FIGURE 9. (a) One of 14 overlapping photos of MIT buildings shot from a street-level tripod. (b) 3D mosaic of street-level photos.

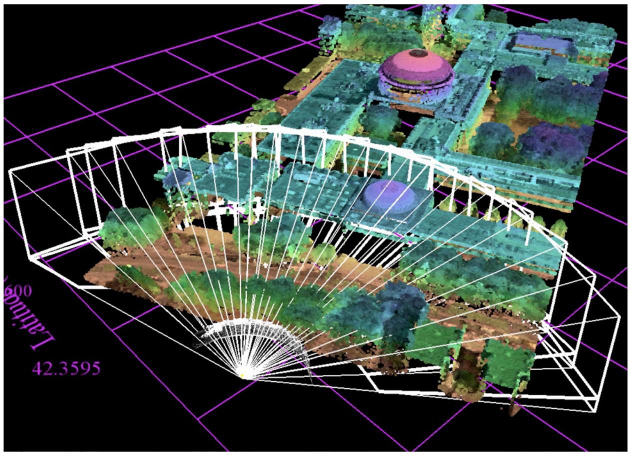


FIGURE 10. Street-level panorama georegistered with the 3D MIT map. [video]

Moreover, high-level knowledge attached to the 3D voxels can propagate into the 2D image planes. Consider, for instance, names of buildings and streets (Figure 12a). Such information is typically available from GIS layers that enter into every standard mapping application currently running on the web. Building and street annotations carry longitude, latitude, and altitude geocoordinates that project onto calibrated photographs via their camera matrices. Urban scene annotations then appear at correct locations within the mosaic (Figure 12b).

Similar labeling of dynamic video clips shot in cities is possible, provided they are georegistered with the 3D map. We follow the same matching procedure for our second street-level background panorama and a co-located foreground video as previously described for our first rooftop example. The ground-level video sequence contains pedestrian and vehicle traffic that have no counterparts in the mosaic. Nevertheless, ray matching successfully aligns the input video to the georegistered panorama (Figure 13a). Building and street names project directly from world space into the moving video stream (Figure 13b). As the movie plays, the annotations track moving image plane locations for urban objects up to residual low-frequency jitter not completely removed by αβγ temporal filtering.

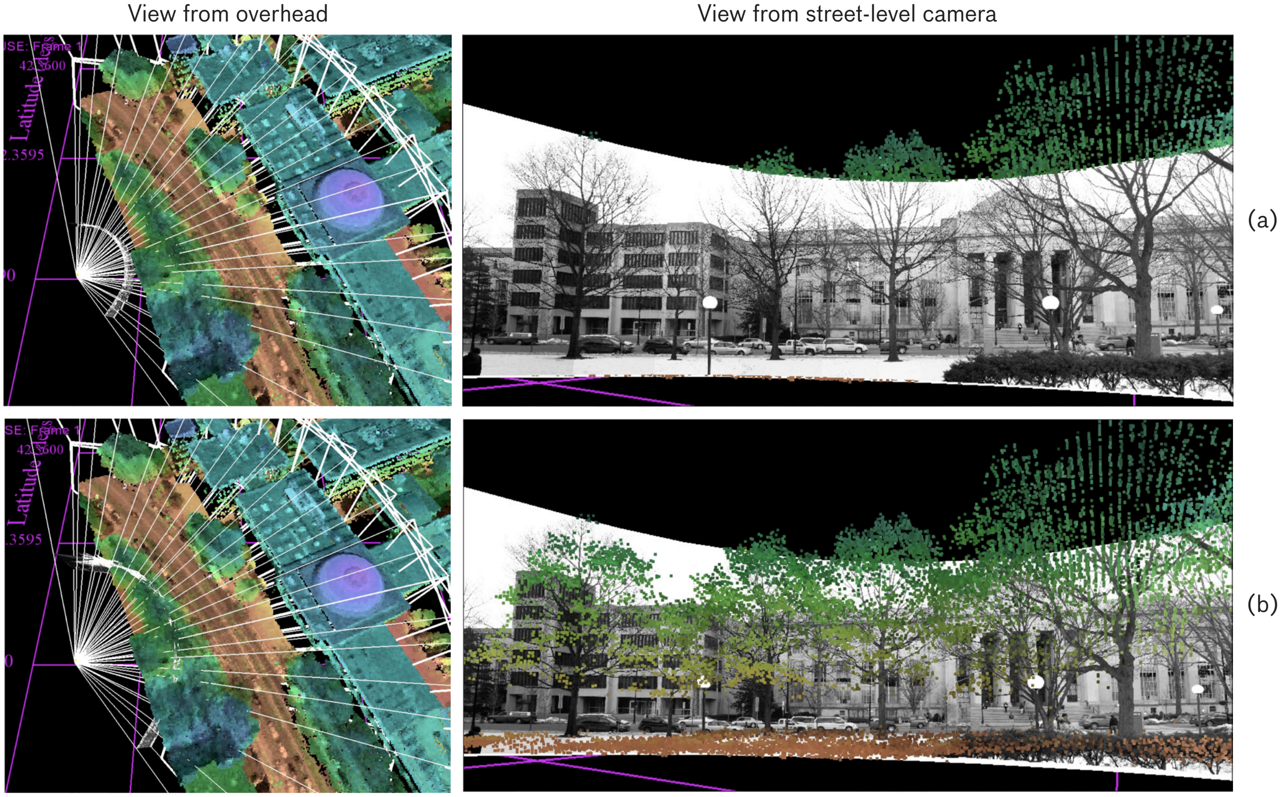


FIGURE 11. (a) Panorama photos occlude camera's view from street-level tripod of ladar point cloud. (b) Ladar points corresponding to summertime tree leaves appear to grow on nude wintertime branches as the range from the camera tripod to the image planes is increased. [video]

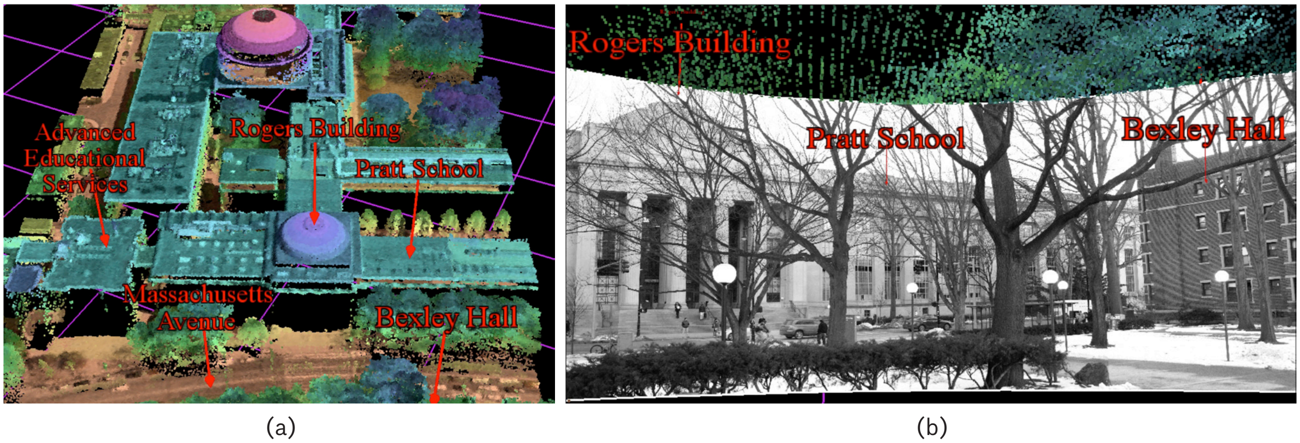


FIGURE 12. (a) Names of buildings and streets appear as annotations in the 3D MIT map. (b) Projected annotations label buildings within the photo mosaic.

This second surveillance example demonstrates that the transfer of abstract information from world space into dynamic image planes is possible for a single camera whose position is fixed in space. Three-dimensional geometry similarly enables knowledge propagation for much more challenging situations involving a moving camera. We therefore now progress from stationary to mobile camera applications.

Rural Reconnaissance via a Single Aerial Camera

Over the past decade, digital cameras have proliferated into many aspects of modern life. A similar, albeit less explosive, growth has also occurred for robotic platforms. To encourage rapid experimentation with both imaging sensors and robots, Lincoln Laboratory held a Technology Challenge called Project Scout in autumn 2010 [18]. The challenge involved remotely characterizing a one-square-kilometer rural area and identifying anomalous activities within its boundary. The challenge further required that solutions developed for this problem had to be economical to implement.

One of many platforms fielded during the 2010 Technology Challenge was a hand-launched UAV. The aerial system's hardware included a Radian sailplane (<$400), a Canon PowerShot camera (<$300), and a Garmin GPS unit (<$100) (Figure 14). The camera and GPS clocks

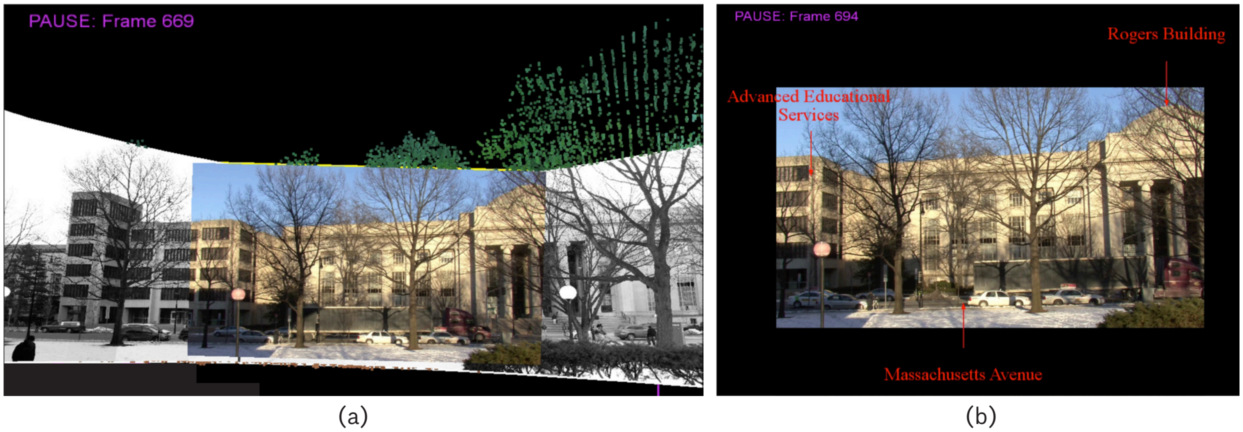


FIGURE 13. (a) One frame from a video sequence automatically aligned with the georegistered mosaic. The dynamic and static imagery were both shot from the same street-level tripod. (b) Annotation labels track stationary buildings and streets (and ignore moving vehicles) within a panning video camera clip. [video a] [video b]

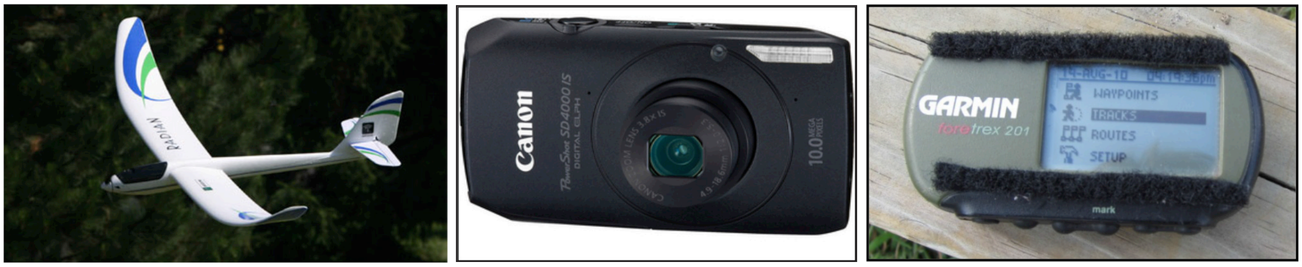


FIGURE 14. Hardware used to collect aerial video over a rural setting included (left to right) a hand-launched sailplane glider, Canon PowerShot camera, and Garmin GPS unit.

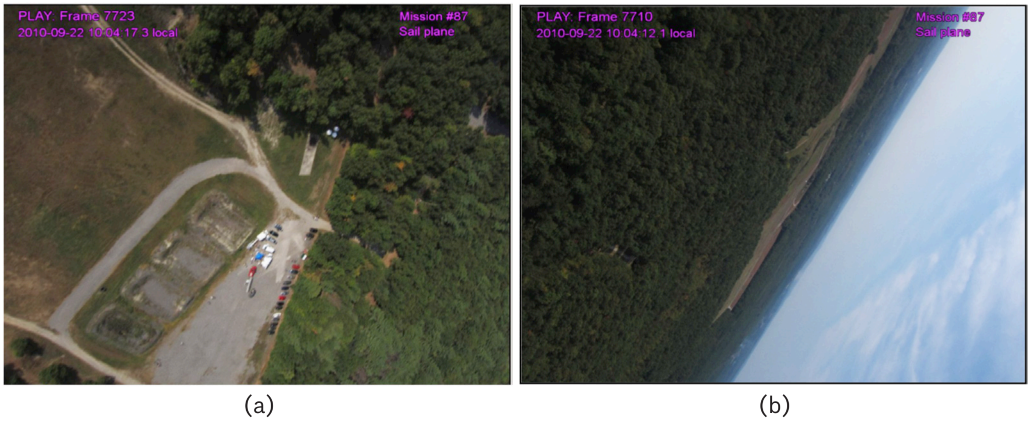


FIGURE 15. Two frames snapped by the digital camera onboard the UAV, which flew up to 430 meters above ground. [video]

were synchronized by taking pictures of the latter with the former. Both sensors were mounted to the UAV's underside prior to launch. Over the course of a typical 15-minute flight, the lightweight digital camera collected a few thousand frames at roughly 3 Hz. When the glider returned from a sortie, the camera's pictures were offloaded from its SD chip and later processed on the ground. Two representative examples of video frames gathered by the aerial vehicle are shown in Figure 15.

Just as for the panoramic MIT photos, the processing pipeline for the aerial video frames begins with SIFT feature extraction and matching. Figure 16 illustrates SIFT tiepoint pairs found for two UAV pictures. Once corresponding features are matched across multiple frames, our system next employs structure-from-motion (SfM) techniques to recover camera poses and sparse scene structure (Figure 17). SfM takes 2D feature matches as input and computes a set of 3D scene points. It also returns relative rotation, position, and focal length parameters for each camera. We employ the Bundler toolkit [19] to solve this highly nontrivial optimization problem [9, 10].

Of the more than 3000 frames passed into the aerial video processing pipeline, approximately 1500 were reconstructed. Given their high computational complexity, the feature extraction and matching steps for this 3D reconstruction were run on Lincoln Laboratory's high-performance, parallel computing system, known as LLGrid [20], with specially parallelized codes [21]. We next applied multiview stereo algorithms developed by Furukawa and Ponce to generate a dense representation for the ground scene [22]. The resulting point cloud of the rural area overflown by the UAV is much more dense than the sparse cloud generated by incremental bundle adjustment.

It is important to recall that conventional digital cameras only capture angle-angle projections of 3D world space onto 2D image planes. In the absence of metadata, digital pictures yield neither absolute lengths nor absolute distances. Therefore, to georegister the reconstructed aerial cameras plus reconstructed point cloud, we must

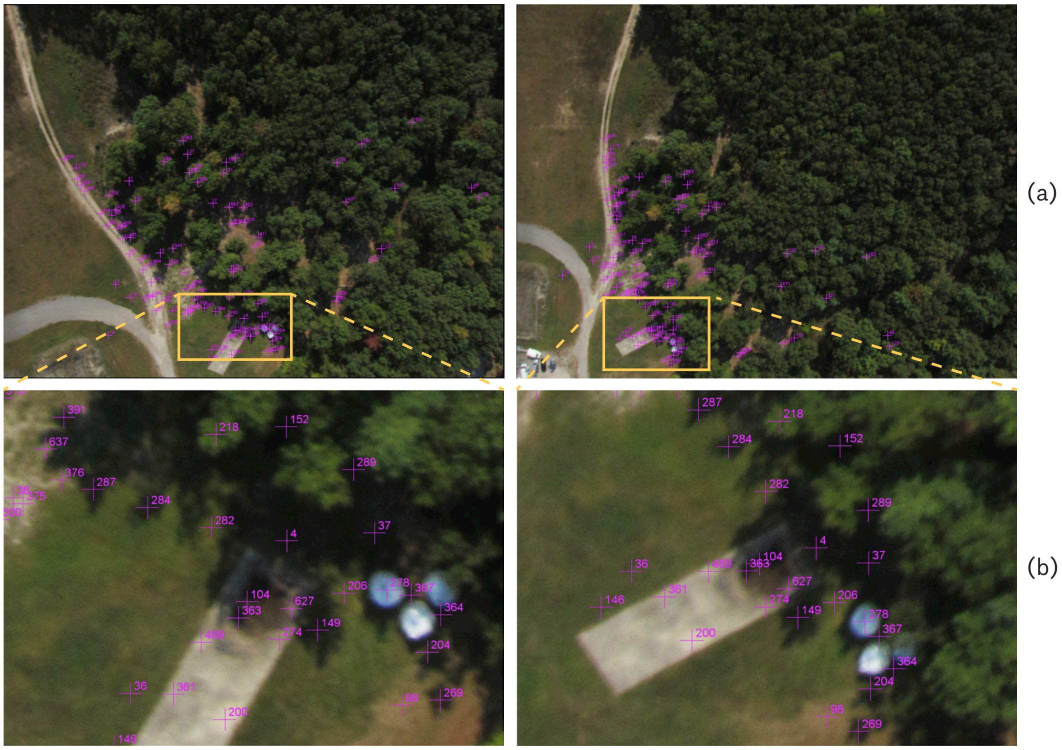


FIGURE 16. (a) 96 SIFT feature matches found between two video frames from the aerial sequence. (b) Zoomed view of tiepoint pairs.

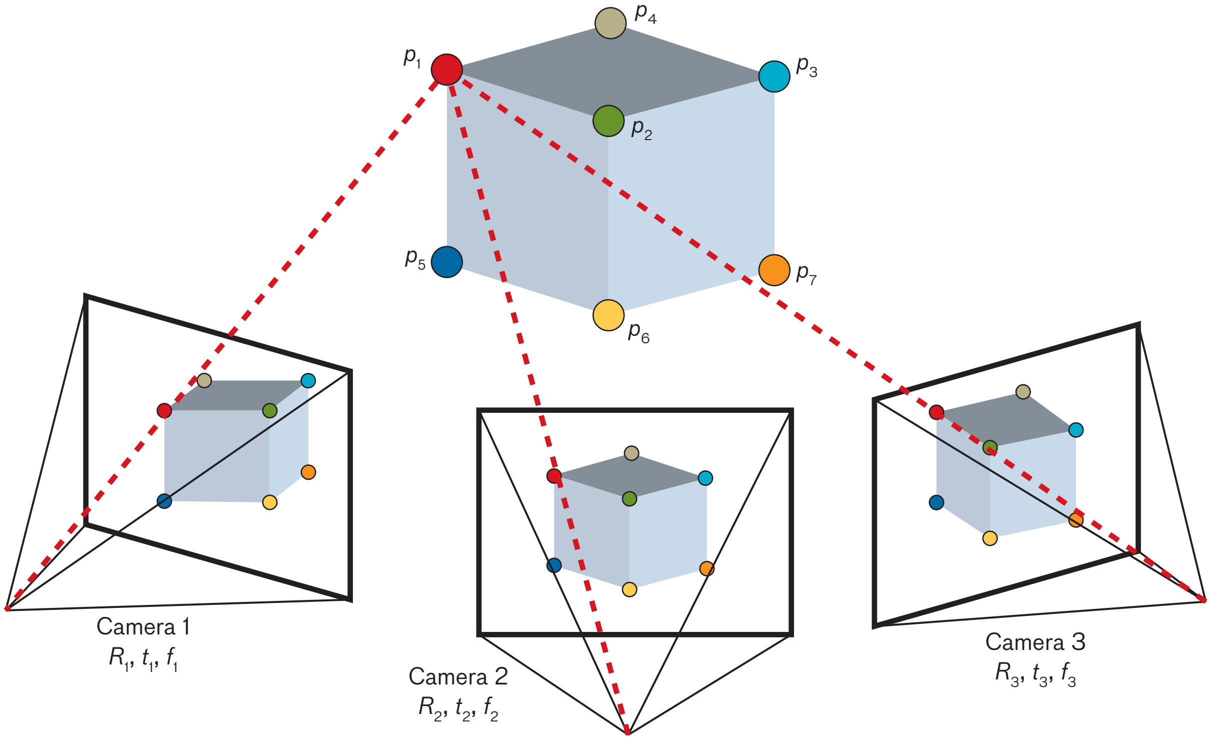


FIGURE 17. Conceptual illustration of structure from motion. Starting from a set of images with calculated tiepoints, one needs to solve for the features' relative 3D locations plus the cameras' rotation, translation, and focal parameters.

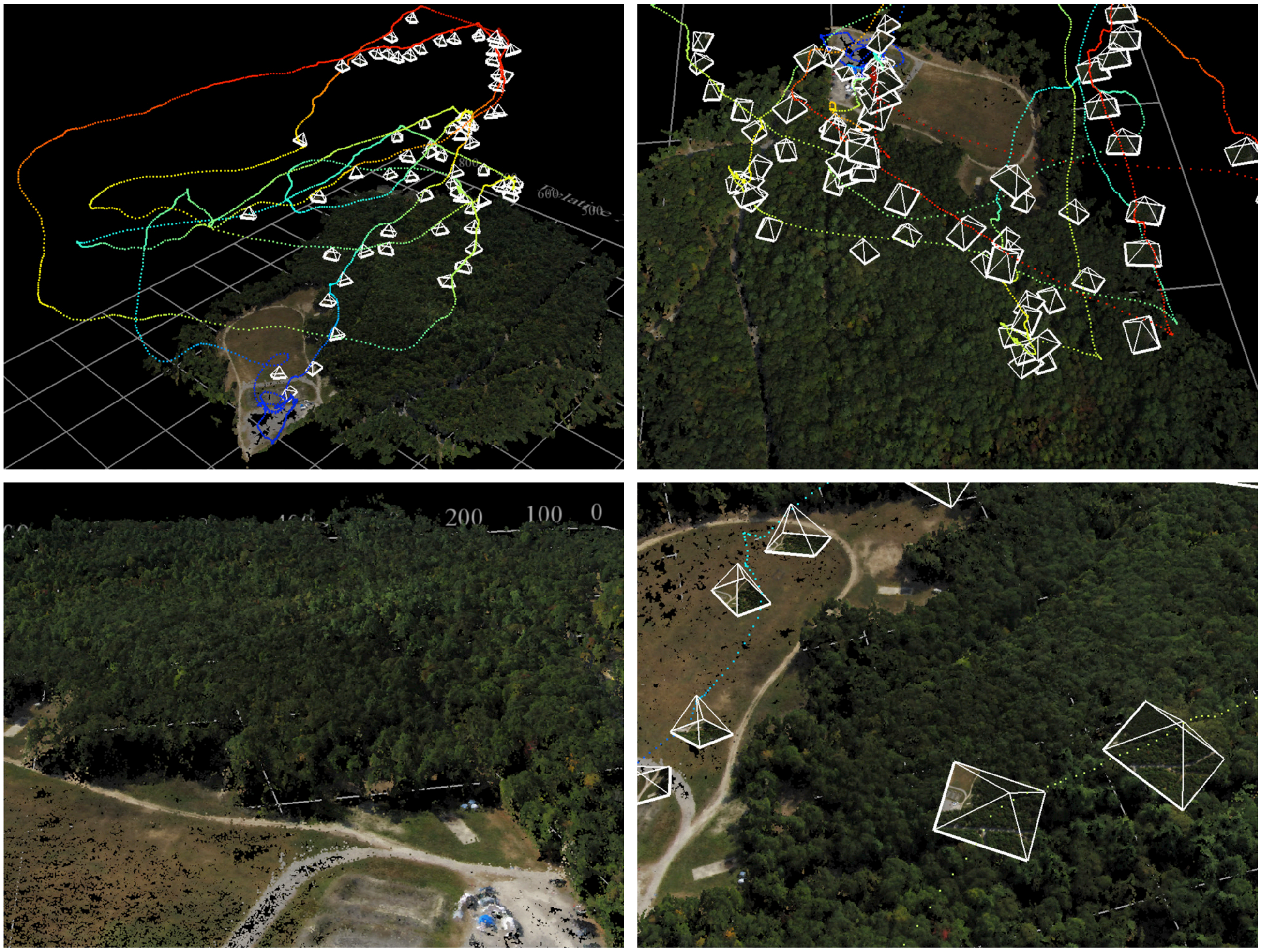


FIGURE 18. Four frames from a movie fly-through of the 3D rural scene densely reconstructed from aerial video frames. Only 74 of approximately 1500 recovered camera frusta are displayed. The curve colored according to height depicts the glider's GPS track. [video]

utilize other sensor data beyond CCD outputs. Unlike in our preceding panorama experiments around MIT, we do not have access to a high-fidelity ladar map for the rural area over which the UAV flew. So we instead exploit measurements from the glider's onboard GPS unit. By fitting the reconstructed flight path to the aerial platform's track, we can derive the global translation, rotation, and scaling needed to lock the relative camera frusta and dense point cloud onto world geocoordinates.

Figure 18 displays geoaligned frusta for the aerial video frames along with the dense terrain map. The glider's GPS track also appears in the figure as a continuous curve colored according to height. After commanding our virtual viewer's camera to assume the same position and orientation as a reconstructed camera's, we may directly compare the alignment between reconstructed aerial frames and the dense point cloud (Figure 19). A human eye must strain to see discrepancies between the 2D and 3D results.

Having recovered 3D geometry from UAV frames, we now demonstrate several examples of aerial video exploitation via geometry, which are difficult to perform with conventional image processing. For instance, video reconstruction plus georegistration yields detailed height maps with absolute altitudes above sea level for ground, water, and trees (Figure 20). The approximate 1-meter ground sampling distance of the measurements displayed in the figure begins to rival those from ladar systems. But the ~$800 cost of our commercial passive imaging hardware for terrain mapping is much less expensive than the cost for active ladars.

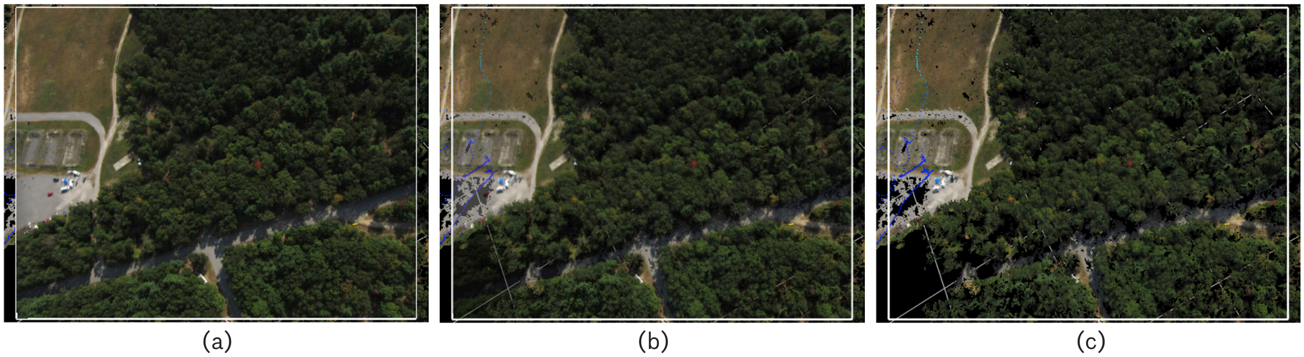


FIGURE 19. One aerial video frame compared against the densely reconstructed 3D point cloud with (a) 0%, (b) 50%, and (c) 100% image plane transparency. [video]

Aerial video orthorectification and stabilization represent further applications of 3D imagery exploitation. The sailplane glider experienced significant jostling during its flight over the rural scene, and its raw video footage looks erratic (Figure 15b). But once the aerial camera's geolocation and geo-orientation are known, it is straightforward to project the Canon PowerShot's reconstructed views onto a ground Z-plane. Figure 21 compares two such orthorectified aerial frames with an orthorectified background image. The discrepancy between the former and latter is estimated to be 2.5 meters. When an entire series of orthorectified aerial frames is played as a movie, the resulting time sequence is automatically stabilized.

As one more example of imagery exploitation, we propagate intelligence from one video frame into another. Once a camera's position and orientation are known, any pixel within its image plane corresponds to a calculable ray in world space. When a user chooses some pixel in a reconstructed UAV picture, we can trace a ray from the selected pixel down toward the dense point cloud. The first voxel in the dense terrain map intercepted by the ray has longitude, latitude, altitude, and range coordinates that its progenitor pixel inherits. Figure 22a illustrates three selected locations in the 44th aerial frame that have been annotated with their associated voxels' ranges and altitudes.

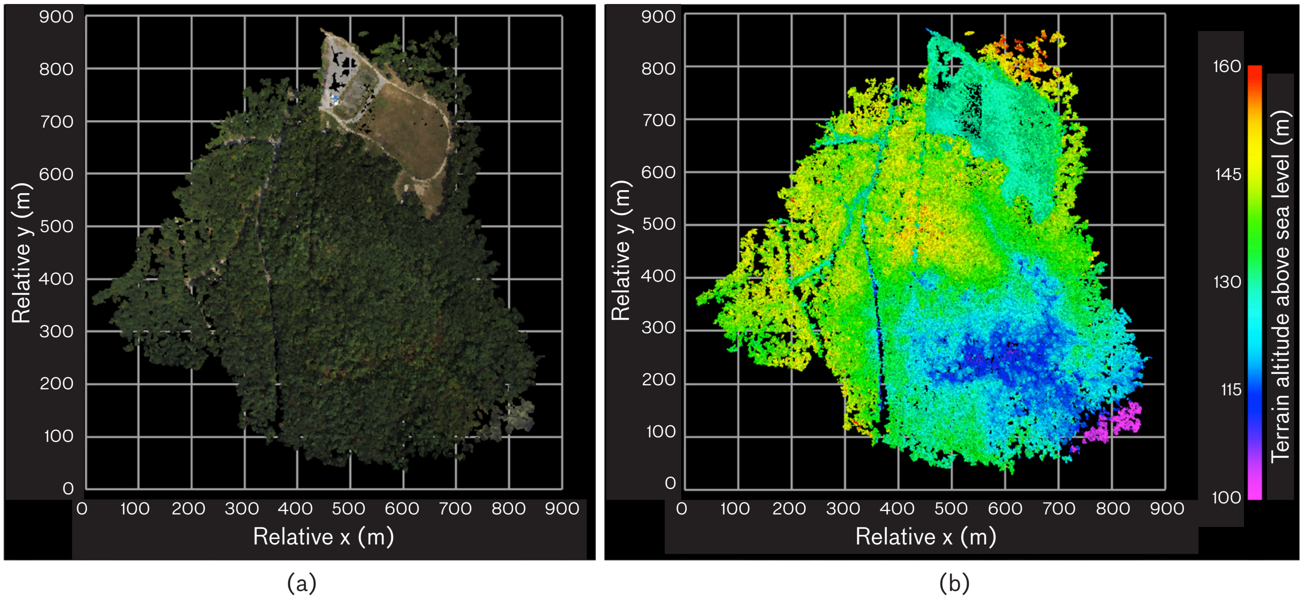


FIGURE 20. Dense point cloud for rural terrain colored according to (a) camera's RGB (red-green-blue) values and (b) height above sea level.

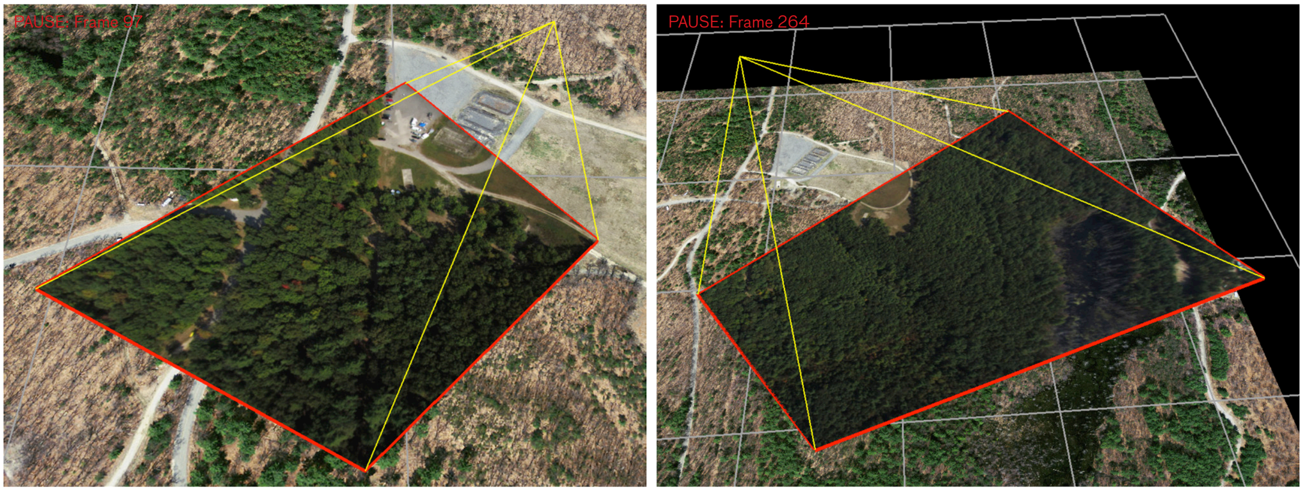


FIGURE 21. Aerial video frames backprojected onto a ground Z-plane and displayed against an orthorectified background image. [video]

One may next inquire which, if any, of the geolocations selected in the 44th video frame reappear in others, such as the 67th in Figure 22b. It is not easy for a human to solve this puzzle by eyeing the two aerial views of the nondescript rural terrain. But a computer can readily deduce the solution. The voxels associated with the pixels in Figure 22a are reprojected into the image plane for the secondary camera. This procedure generates the single tiepoint match in Figure 22c. This example illustrates how information can propagate from one 2D view into another when 3D geometry acts as a mathematical conduit.

Working with thousands of video frames gathered by a single mobile camera, we have demonstrated the imagery reconstruction, imagery georegistration, and intelligence propagation algorithms diagrammed in Figure 2. The same computer vision techniques may be applied to the more difficult problem of exploiting digital pictures gathered by multiple cameras with a priori unknown intrinsic and extrinsic parameters. We therefore now move from the second to third application depicted in Figure 3.

Urban Mapping via Several Semicooperative Ground Cameras

In summer 2009, a small team of Lincoln Laboratory volunteers set out to collect a large, urban photo set for 3D exploitation purposes. MIT's main campus was selected as a surrogate small city because the natives would not likely be perturbed by unorthodox data-gathering techniques. The volunteers ran around MIT shooting as many digital photos as possible with a variety of cameras. During the first five minutes of the first photo shoot, pictures were selected with care and precision. But as time passed, choosiness went down while collection rate went up. Over the course of five field trips, more than 30,000 stills were collected around MIT.

Recovering geometric structure from 30,000+ images is akin to solving a complex jigsaw puzzle. Most of the pictures were shot outdoors. But a few thousand photos were intentionally taken inside some MIT buildings with the hope of connecting together exterior and interior views. All of the photos were collected inside urban canyons where the scene changed significantly with every few steps. The photo set's diversity can be seen among the representative examples pictured in Figure 23.

Just as for the aerial frames collected over the rural scene, the processing pipeline for the 30,000+ MIT ground photos begins with feature extraction and matching. SIFT matching imposes an initial topological ordering upon the input set of quasi-random photos. Each image may be regarded as a node in a graph whose edges indicate feature pairings. Figure 24 visualizes this abstract network for the 30,000+ photos. The interactive graph viewer appearing in this figure was developed by Michael Yee [23]. It allows a user to navigate through large imagery collections. With Yee's graph tool, one can develop a global understanding of a SIFT graph's content as well as drill down to inspect individual pictures of interest.

Once SIFT features have been matched between multiple views, a machine can recover geometry information via incremental bundle adjustment [9, 10, 21]. Figure 25 displays reconstruction results for 2300+ of the 30,000+ pho-

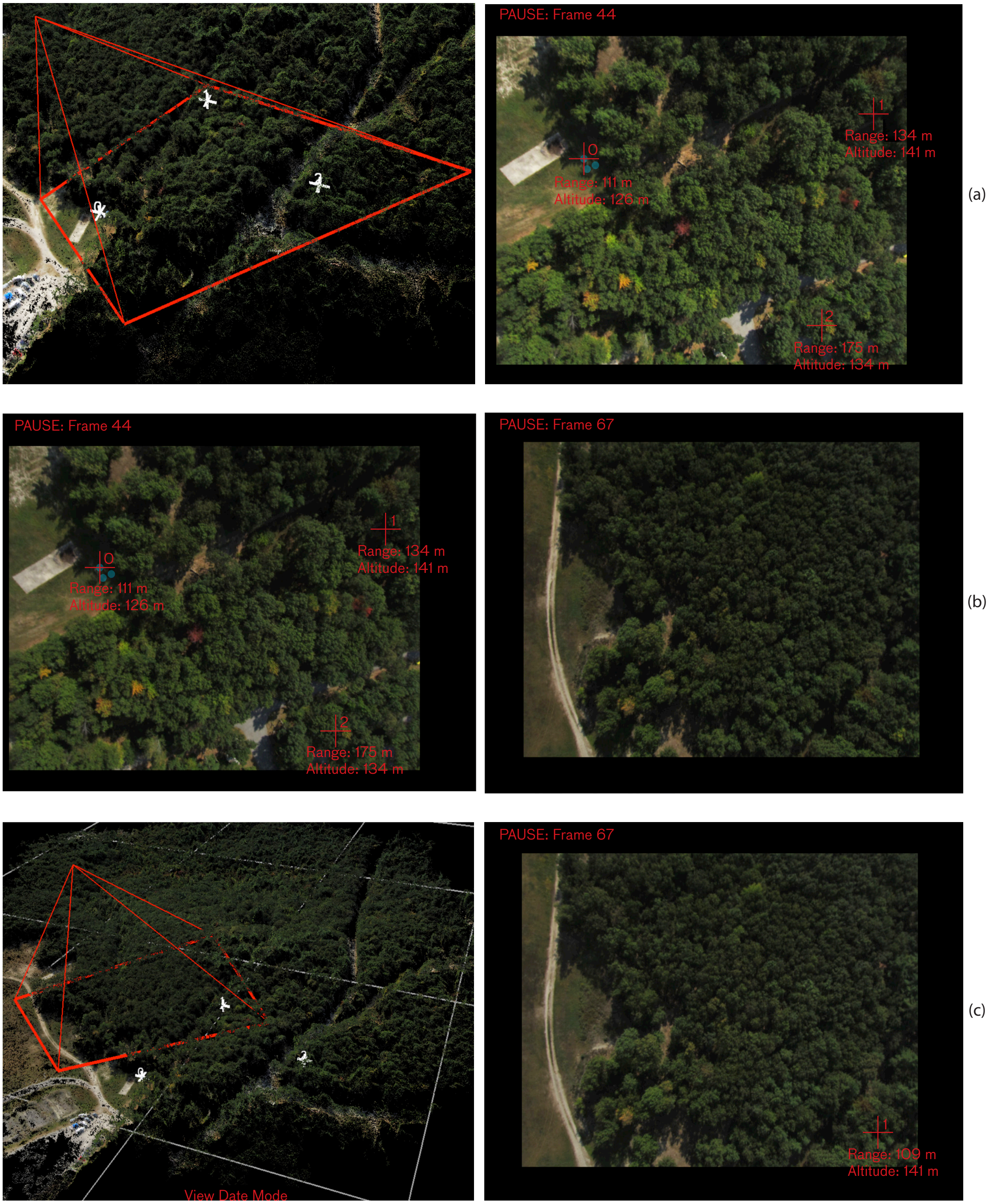


FIGURE 22. (a) 2D pixels selected within the video frame on the right correspond to world-space rays on the left. The selected points are annotated with ranges and altitudes coming from voxels in the dense point cloud that the rays intercept. (b) Determining which, if any, of the pixels selected in the 44th video frame appear in the 67th is difficult for a human eye. (c) Geometry reveals that only pixel 1 in the 44th video frame has a visible counterpart in the 67th frame.

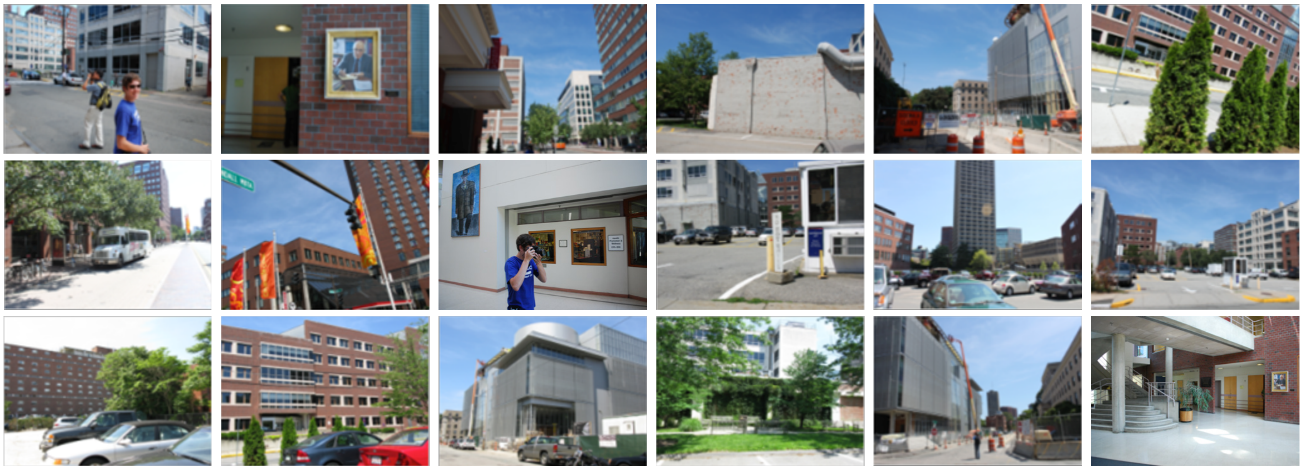


FIGURE 23. Eighteen representative photos from 30,000+ shot semi-randomly around MIT in summer 2009.

tos. The colored point cloud in Figure 25a depicts the relative 3D shapes of several buildings located on MIT's eastern campus. When we zoom in, reconstructed photos are represented as frusta embedded within the point cloud (Figure 25b). The snapshots in Figure 25 are actually two frames from a movie sequence in which a virtual camera flies through the 3D scene. By viewing the entire movie, one starts to gain an intuitive feel for MIT's complex urban environment.

In addition to the set of 30,000+ ground photos, we also have access to orthorectified aerial imagery and georegistered ladar data collected over MIT. The former is publicly available from the MassGIS website [24], while the latter was obtained from the Topographic Engineering Center of the U.S. Army's Engineer Research and Development Center. Figure 26 exhibits representative samples of these 2D and 3D images.

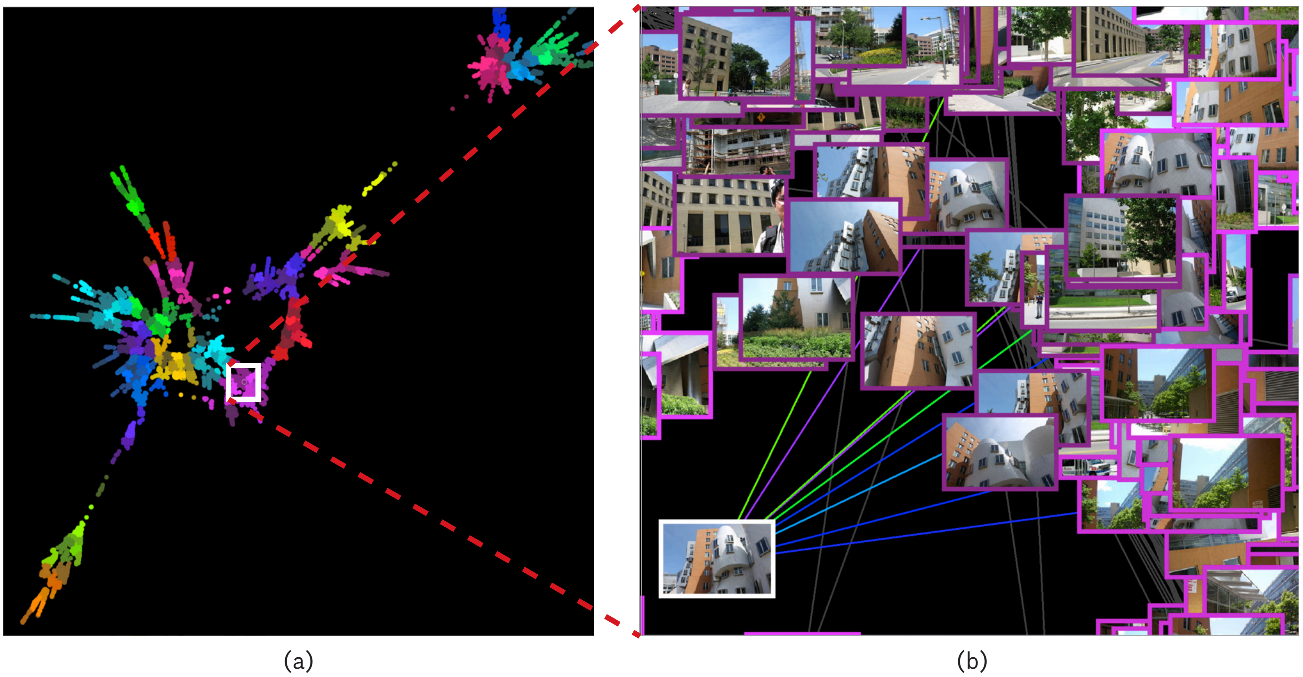


FIGURE 24. SIFT graph for MIT ground photos. (a) Nodes within the graph representing images are hierarchically clustered and colored to reveal communities of similar-looking photos. (b) Nodes automatically turn into image thumbnails when the graph viewer is zoomed in. [video]

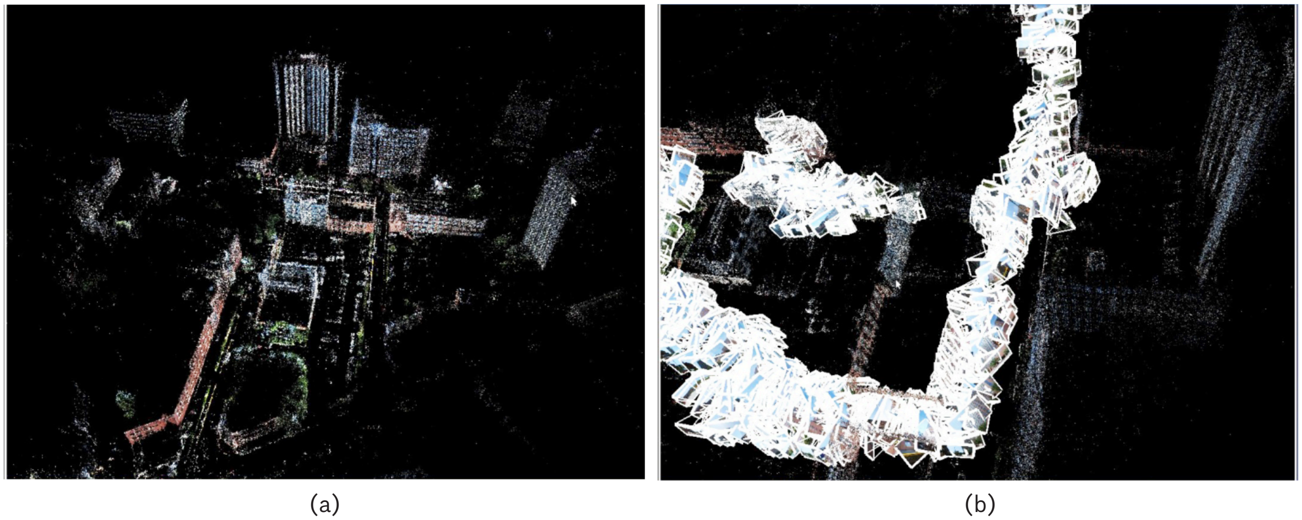


FIGURE 25. Incremental bundle adjustment results for 2317 ground photos shot around eastern MIT. (a) Reconstructed point cloud illustrates static urban 3D structure. (b) White-colored frusta depict relative position and orientation for photos' cameras. [video]

Photos and ladar point clouds represent low-level data products that contain millions of pixels and voxels. For data-fusion purposes, it is more useful to work with higher-level models, which abstract out geometrical invariants common to all views. We consequently developed a semiautomatic method for constructing building models from the 2D and 3D inputs.

We first manually extract footprints from the orthorectified aerial imagery (Figure 26a). Each footprint corresponds to some part of a building with approximately constant height. Judgment is exercised as to a reasonable level-of-detail for city structure contours. After 2D footprints are drawn, a computer extrudes them in the Z direction by using ladar data to determine absolute heights above sea level. The resulting prisms capture basic shape information for individual buildings (Figure 26c).

We applied this semiautomatic modeling procedure to 29 buildings around MIT. The models appear superposed against the ladar point cloud in Figure 27. It is worth noting that the ground surface for this part of Cambridge, Mass., is well represented by a plane positioned 2.5 meters above sea level. Though this ground plane may also be simply modeled by a geometrical primitive, it is not displayed in the figure for clarity's sake.

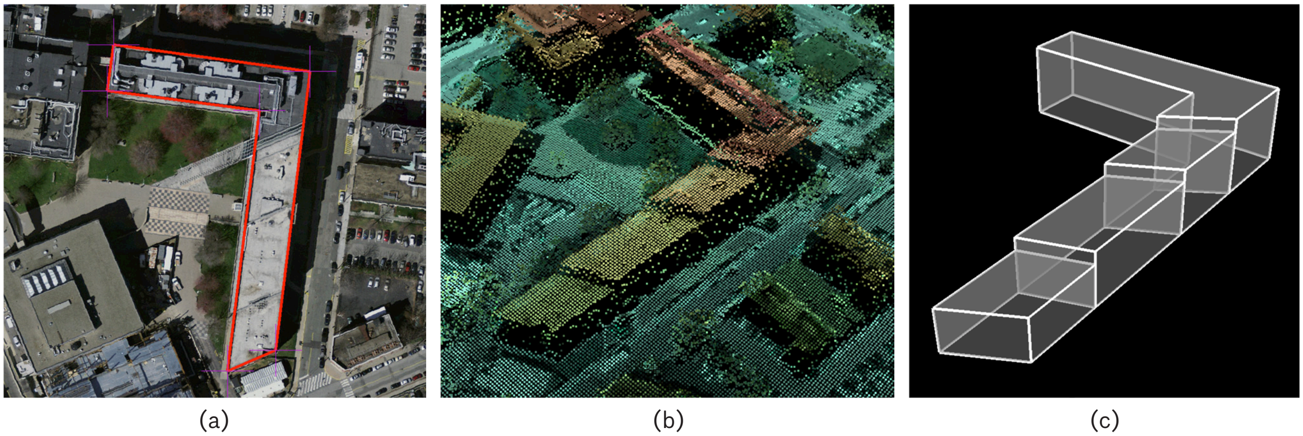


FIGURE 26. Semiautomatic construction of urban building models. (a) Manually selected corners establish a footprint for one particular building. (b) Ladar data supply building height information. (c) A model capturing a building's gross shape is generated automatically from the 2D footprint and 3D heights for Z-plane surfaces.

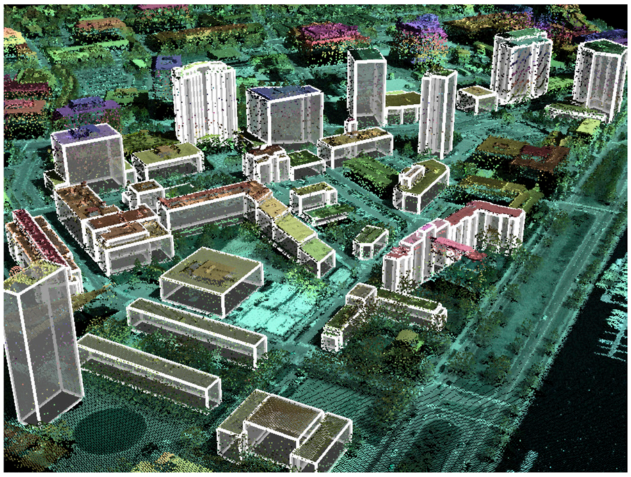


FIGURE 27. Models for 29 buildings around MIT's eastern campus are superposed against the ladar point cloud for Cambridge, Mass.

The 3D building models establish a background map onto which the reconstructed ground photos may be georegistered. Specifically, tiepoint correspondences were manually established between pixels in 10 selected photos and counterpart corner points among the building models. The tiepoint pairings were subsequently used to set up and solve a system of equations for the global translation, rotation, and scaling needed to geoalign all the reconstructed photos with the 3D map [25].

The georegistered ground photos reside among the MIT building models in Figure 28. For clarity, only 230 of the 2300+ reconstructed cameras' frusta are displayed in the figure. Looking at their pattern in Figure 28a, one can discern the path followed by the Lincoln Laboratory adventurers as they roamed around the institute. In the zoomed view of Figure 28b, the ground photos are visible as oriented image planes, while their cameras' geopositions are indicated by frusta apexes.

The screenshots in Figure 28 (as well as many other figures in this article) were taken from a 3D viewer based upon the OpenSceneGraph toolkit [26]. With this viewer, a user may select an individual frustum for inspection by mouse-clicking near its apex. The viewer's virtual camera then flies into the selected frustum's position and assumes its 3D orientation. The user can also modulate the transparency level for the selected frustum's image plane. As Figure 29 demonstrates, fading away the foreground photo enables comparison with the background building models. Though the georegistration of the 2300+ ground photos with the building models exhibits small errors primarily caused by imperfect camera pointing angles, the alignment between thousands of semi-randomly shot ground photos and the abstract map derived from the aerial data is striking.

After urban photos have been geoaligned with the 3D map, automatic segmentation of their building contents becomes tractable. For instance, consider the two pictures in Figure 30a. The building facades in these images have positions, orientations, and colorings that are a priori unknown. Consequently, training a classifier to separate out individual buildings from these views is

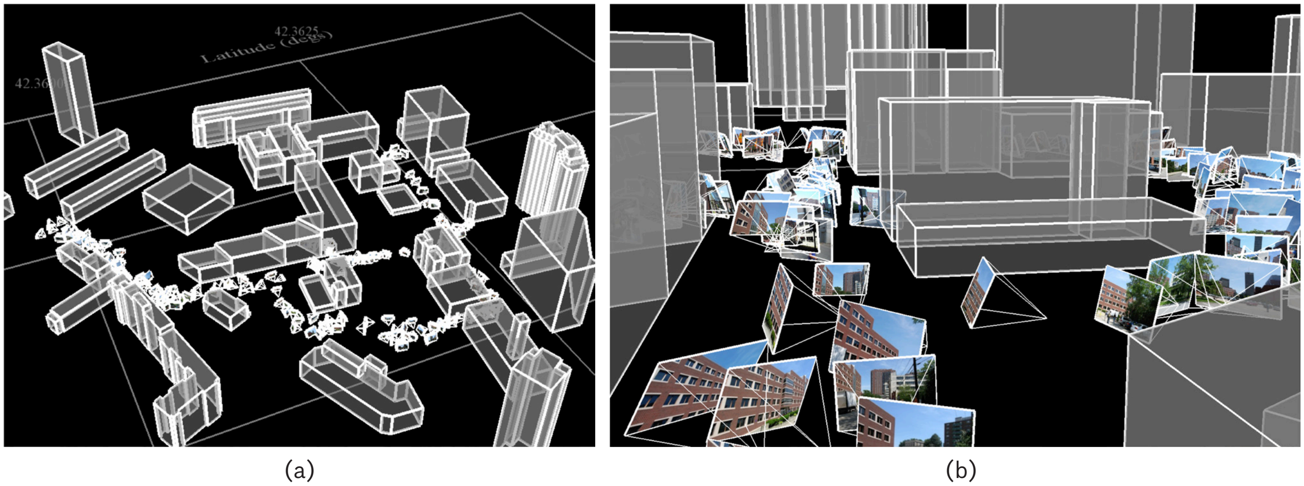


FIGURE 28. Georegistered ground photos displayed among 3D building models. (a) 230 of 2317 reconstructed photos represented as frusta. (b) Close up view of frusta illustrates cameras' geopositions and pointing directions. [video]

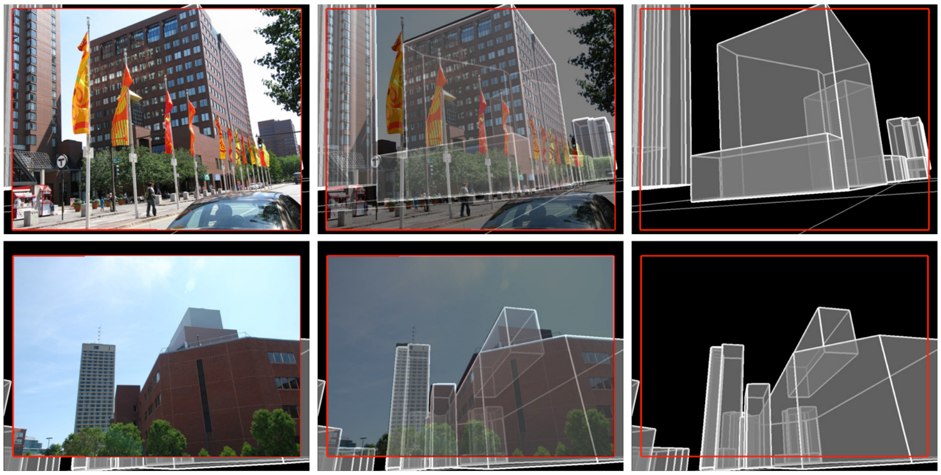


FIGURE 29. The alignment between 2D ground photos and 3D building models can be seen after the 3D viewer's virtual camera assumes the same position and orientation as the georegistered cameras and their image planes are faded away. [video]

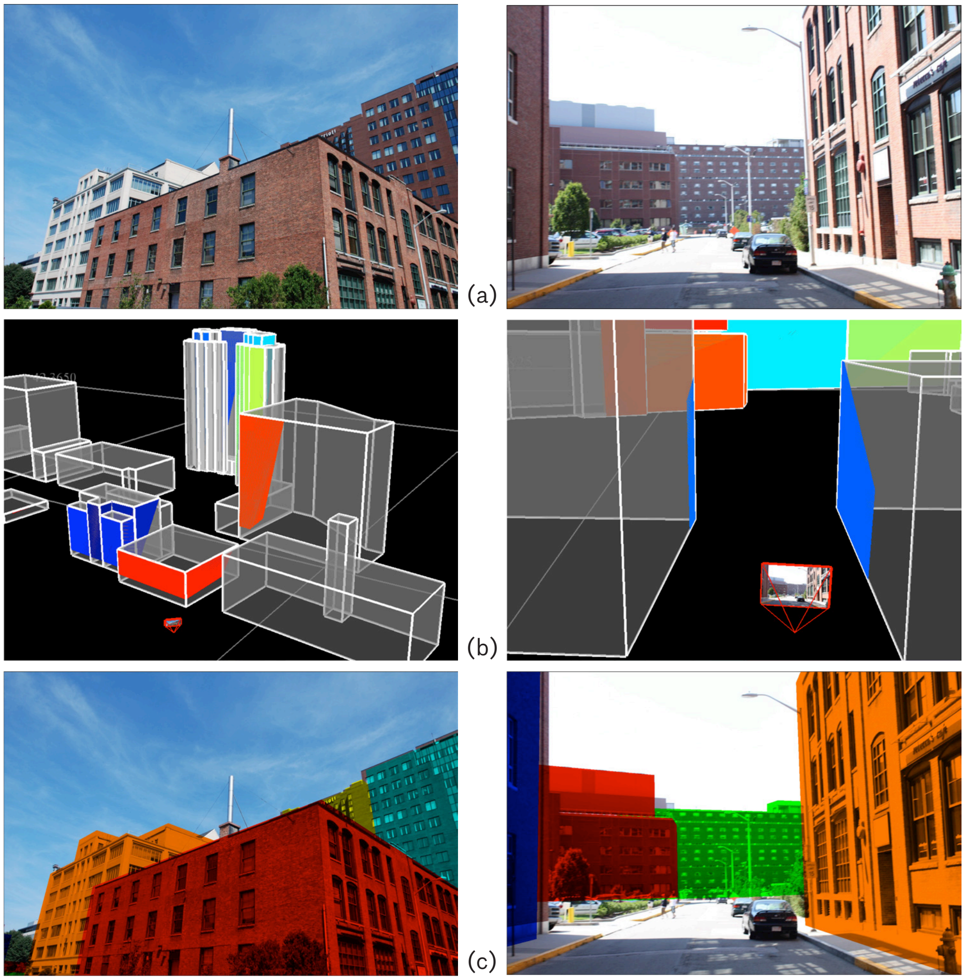
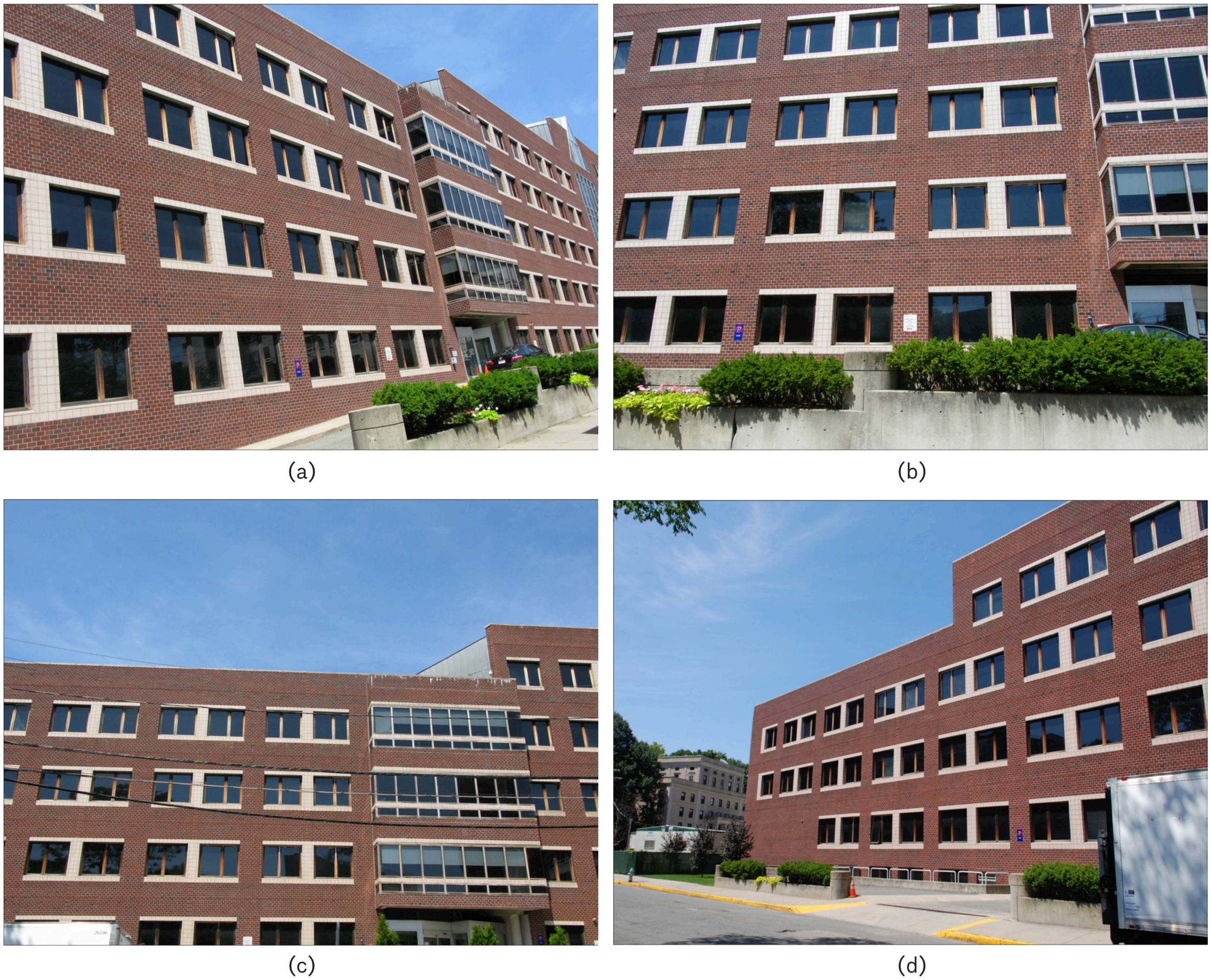


FIGURE 30. Building segmentation within street-level imagery. (a) Two representative photos shot around MIT. (b) Colored polygons indicate clipped 3D wall regions that are visible within the photos' georegistered frusta. (c) Image-plane pixels corresponding to individual buildings are tinted with distinct colors.

difficult. But once geometry relationships between image planes and the abstract map are known, a machine can identify and clip 3D polygons that are visible to the reconstructed cameras (Figure 30b). When the clipped world-space polygons are projected onto the 2D image planes and colored according to model identity, building masks are effectively generated for the photos (Figure 30c).

Geoaligned ground photos may further be exploited to develop photorealistic 3D maps for complex city scenes. This ambitious objective represents an active area of academic and commercial research. Popular 3D modeling programs such as SketchUp provide interactive tools for constructing urban buildings [27]. But texturing building facades with digital photo content remains a manually intensive process. Therefore, it is instructive to investigate how thousands of reconstructed images could be used to semiautomatically paint details onto relatively coarse 3D models.

Given a set of digital pictures such as those of MIT's medical center in Figure 31, a machine can identify rectangular faces of world-space models onto which they backproject. The 3D corners for each building face are next converted into 2D planar coordinates, and the corner points are projected into the photos' image planes. Geometrical relationships between the former and latter planes define homographies that can be used to orthorectify building facades. Orthorectified "decals" are generated by applying the homographies to all facade pixels within the original images. Figure 32 exhibits building facade decals corresponding to the photos in Figure 31.



(a)

(b)

(c)

(d)

FIGURE 31. Four different photos of MIT's medical building.

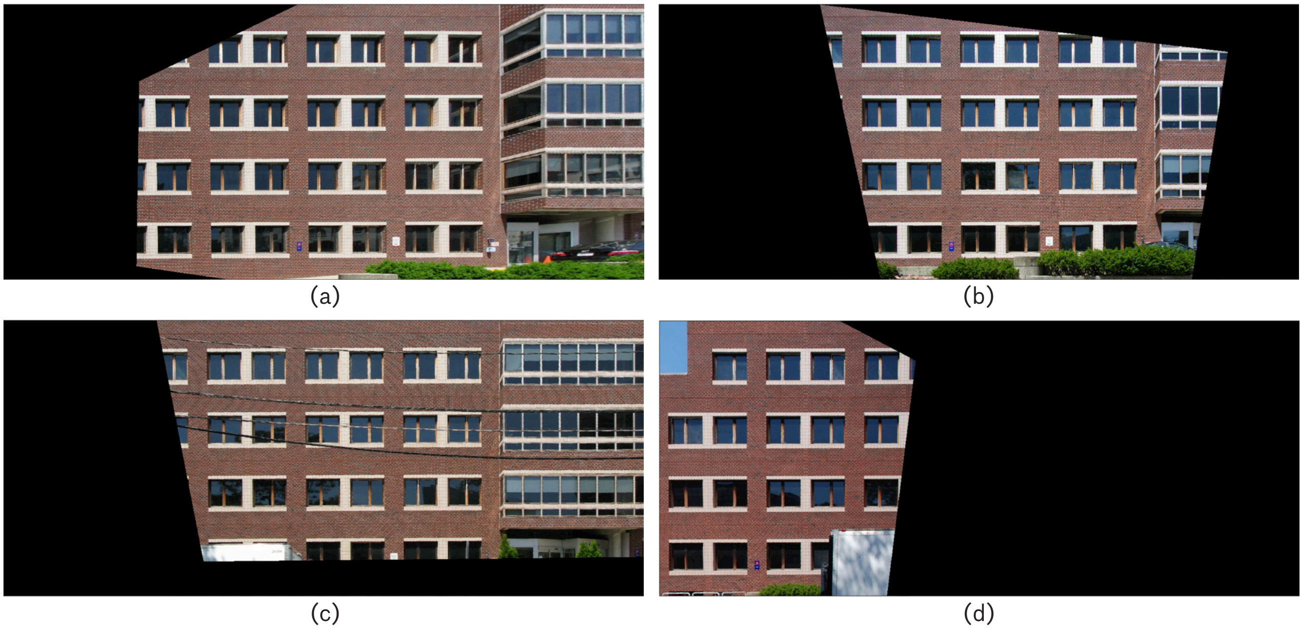


FIGURE 32. The four views in Figure 31 transform into medical building decals after model-induced orthorectification.

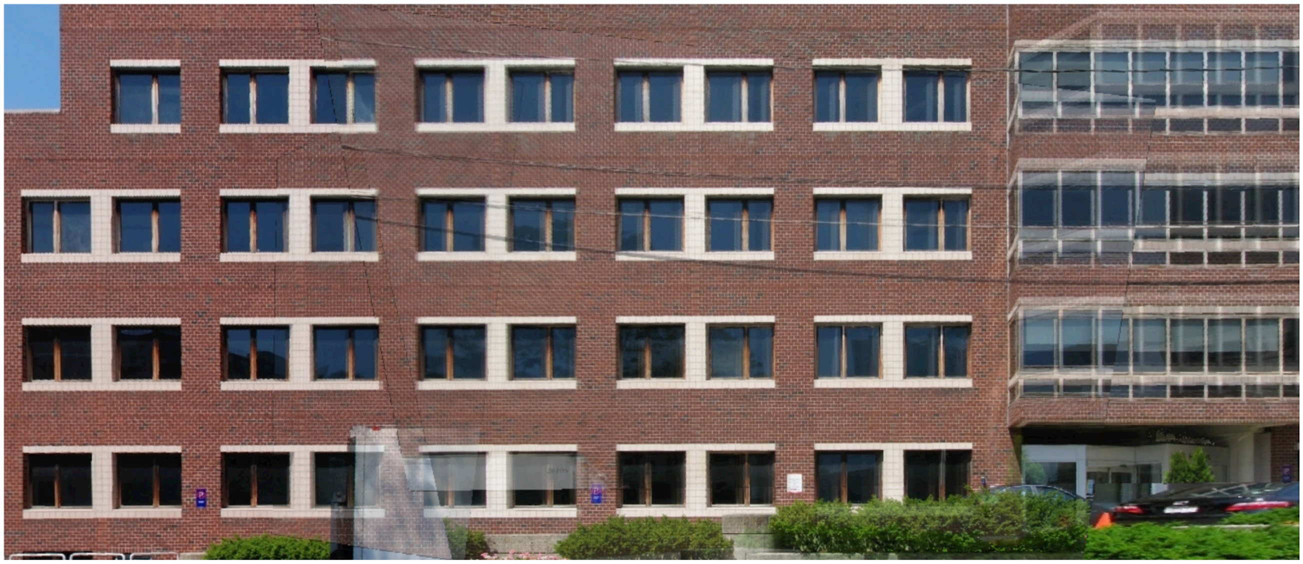


FIGURE 33. Simple RGB color averaging of the orthorectified decals in Figure 32 yields this composite mosaic.

Because the decals for any particular building facade all reside in the same planar coordinate system, they may be mosaiced together to generate a composite that covers an entire building wall. We have made no attempt to implement sophisticated color averaging or outlier detection. Instead, we simply average together any non-null pixels to compute RGB values inside the mosaic. Figure 33 illustrates the composite decal for the medical center's facade.

We have applied this orthorectification and mosaicing procedure to four georegistered ground photos for three different walls among our 29 building models. The final composite decals appear as 3D textures inside the MIT map (Figure 34). The results look visually appealing, and with more work could be extended to other buildings. Alternatively, orthorectified decals could be used as starting points for refining building models to incorpo-

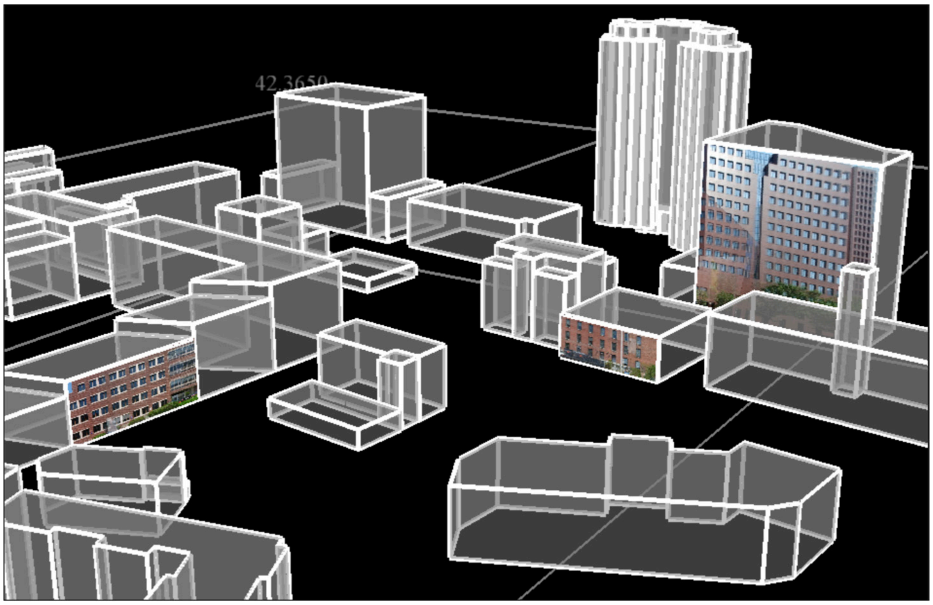
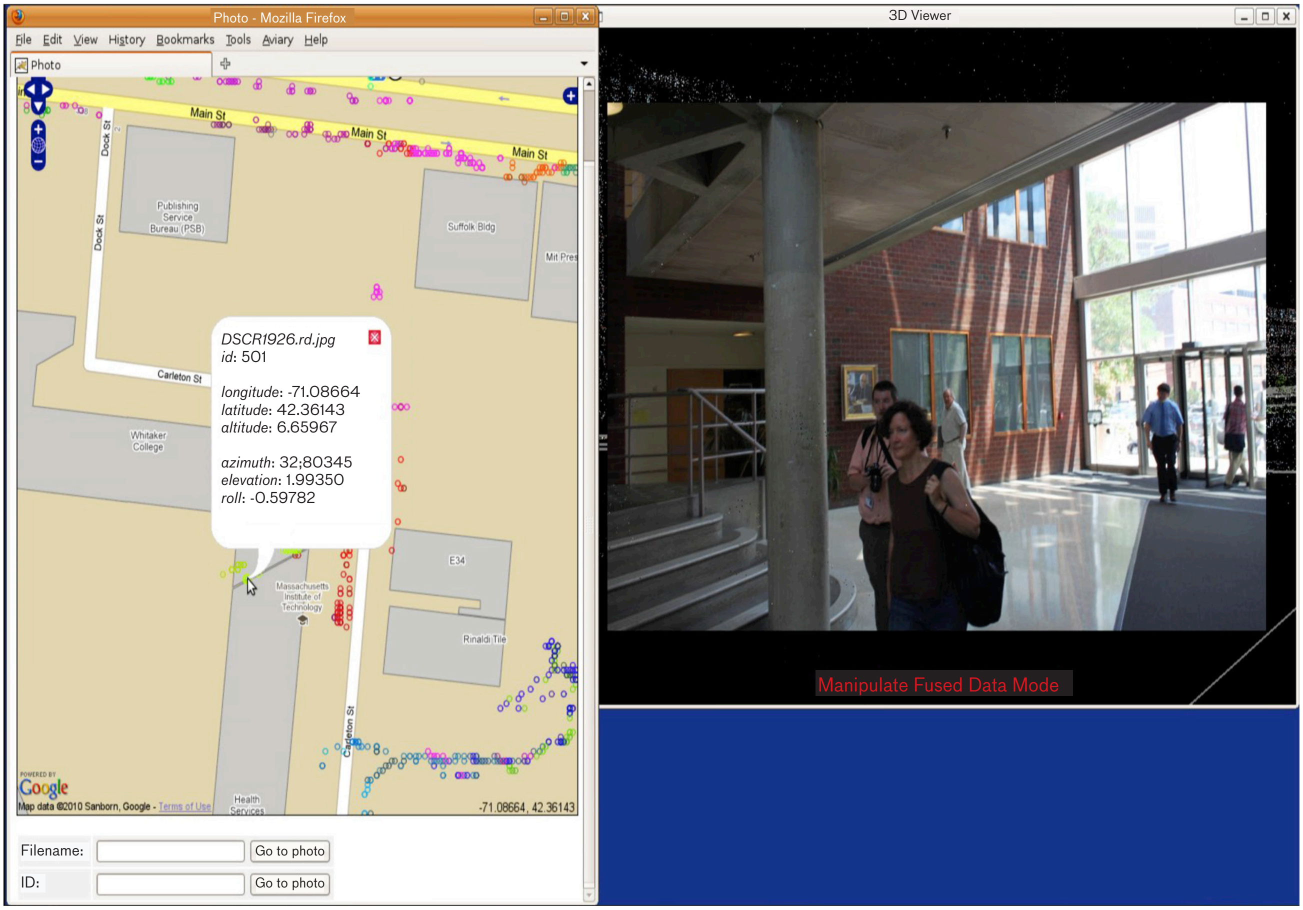


FIGURE 34. Mosaic decals for three building walls, generated from 12 reconstructed ground photos, are textured onto 3D models.



DSCR1926.rd.jpg
id: 501

longitude: -71.08664
latitude: 42.36143
altitude: 6.65967

azimuth: 32;80345
elevation: 1.99350
roll: -0.59782

Filename: Go to photo

ID: Go to photo

FIGURE 35. Synchronized Google map and 3D viewer displays of reconstructed ground photos. Camera geolocation and geo-orientation information are displayed within the web browser when a user clicks on a colored dot. The Google map interface was developed by Jennifer Drexler. [video]

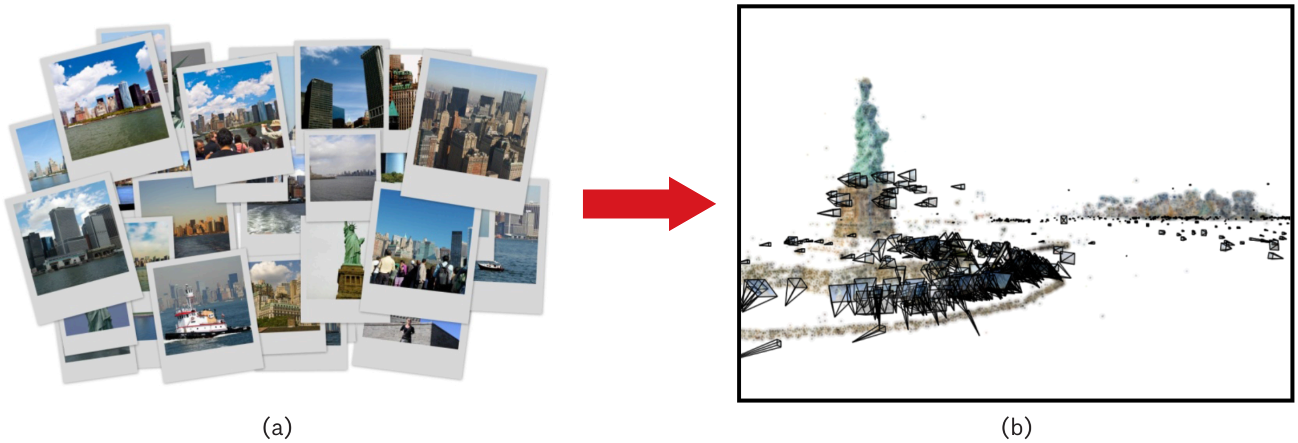


FIGURE 36. Random Flickr photos of the downtown NYC skyline and Statue of Liberty become geometrically organized following 3D reconstruction. [video]

rate geometric details like window and door locations. We leave such model refinement to future work.

The digital pictures shot around MIT during summer 2009 were densely collected in many quasi-random pointing directions. As a result, the georegistered photos form a complicated mess. Trying to pick out a single frustum from among thousands within a 3D viewer is not simple. We thus combined our OpenSceneGraph display tool with a web browser interface in order to simplify human interaction with the rich dataset. Figure 35 displays a Google map of MIT's campus onto which the 2300+ reconstructed photos are overlaid as colored dots. When the user clicks on an individual dot, the message is sent from the web browser to the 3D viewer that commands its virtual camera to assume the position and pointing of the reconstructed camera. The user may then view the selected photo inside the 3D map.

The particular picture appearing in Figure 35 is noteworthy because it was obviously taken indoors. This photo's setting— MIT's medical center—has large glass walls. Consequently, some images snapped inside the medical building share SIFT feature overlap with others shot outside. The results in Figure 35 thus constitute an existence proof that geocoordinates for cameras located inside GPS-denied environments can be derived via computer vision.

Social Media Mining via Many Uncooperative Cameras

The examples of 3D imagery exploitation presented in the preceding sections have involved progressively greater a priori camera uncertainty that requires increasingly greater data processing to resolve. We now turn to the social media mining application depicted at the far right in Figure 3, working with Internet pictures that have little or no useful accompanying metadata. Such uncooperatively collected imagery looks significantly more heterogeneous than the datasets considered so far. Nevertheless, the basic algorithm flow in Figure 2 may be applied to exploit thousands of digital pictures harvested from the World Wide Web.

We begin by downloading more than 1000 photos of the lower Manhattan skyline and the Statue of Liberty from the Flickr website [28]. This photo-sharing site contains vast numbers of pictures that users have tagged as generally related to New York City (NYC). But our initial dataset is otherwise unorganized (Figure 36a).

Just as for all the preceding imagery exploitation examples, processing of the NYC photos begins with SIFT feature extraction and matching. As was done for the rural aerial and urban ground pictures shot by mobile cameras, we recover 3D structure for the 1000+ Flickr photos via the SfM approach of Snavely et al. [9, 10, 21]. Relative camera positions along with Statue and skyline geometry are illustrated in Figure 36b. These results required four hours to generate on LLGrid.

In order to georegister the Flickr photo reconstruction to a longitude-latitude grid, we again need data beyond just imagery pixels. So we construct a 3D map for NYC, starting with aerial ladar points. In particular, we work with a Rapid Terrain Visualization (RTV) map collected on 15 October 2001 (Figure 37a). These data have a 1-meter

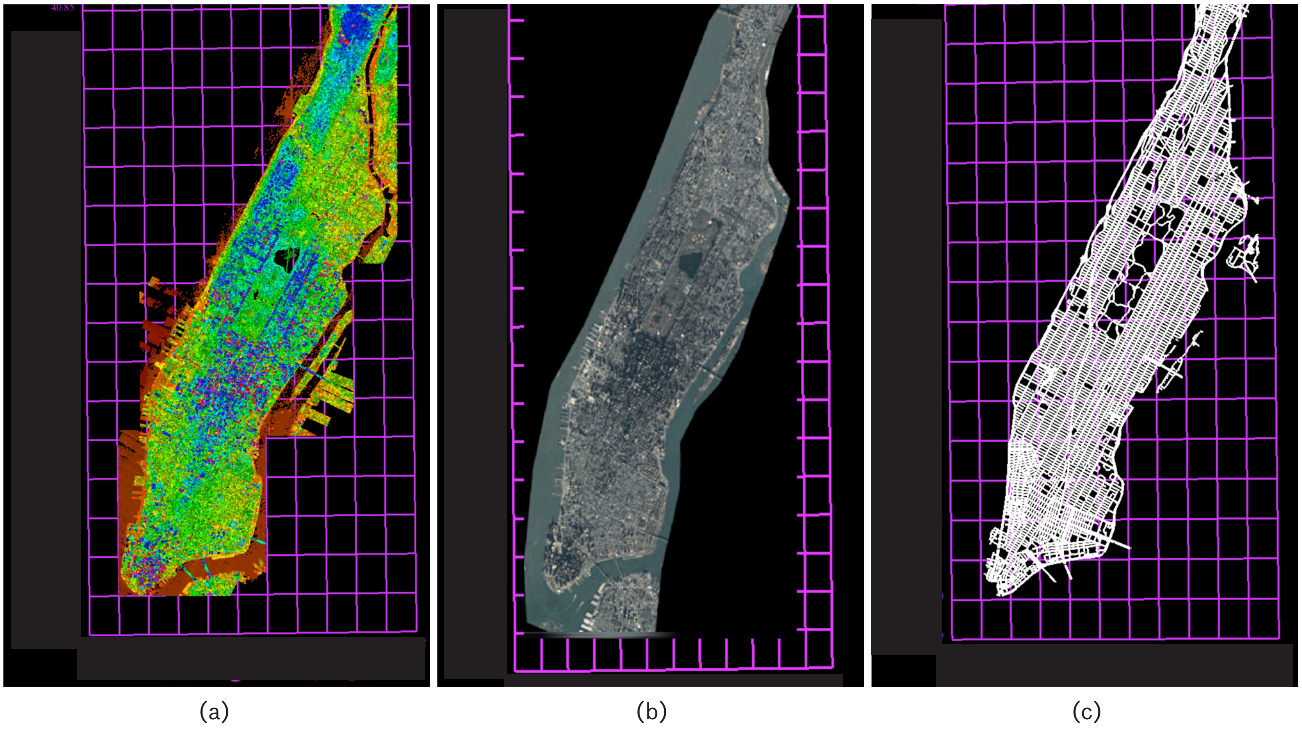


FIGURE 37. 3D NYC map ingredients. (a) Ladar map colored according to height. (b) Satellite image. (c) GIS layer representing NYC's road network.

ground sampling distance. By comparing geolocations for landmarks in this 3D point cloud with their counterparts in other geospatial databases, we estimate that the ladar data have a maximum local georegistration error of 2 meters.

Complex urban environments are only partially characterized by their geometry. They also exhibit a rich pattern of intensities, reflectivities, and colors. Therefore, we next fuse an overhead image with the ladar point cloud. Specifically, we work with Quickbird satellite imagery that covers the same area of NYC as the RTV data (Figure 37b). Its 0.8-meter ground sampling distance is comparable to that of the ladar imagery.

We also introduce GIS layers into the urban map (Figure 37c). Such layers include points (e.g., landmarks), curves (e.g., transportation routes), and regions (e.g., political zones). GIS databases generally store longitude and latitude coordinates for these geometrical structures, but most do not contain altitude information. Fortunately, height values can be extracted from the ladar underlay once lateral GIS geocoordinates are specified.

After combining together the ladar points, satellite image, and GIS data, we derive the 3D map of NYC presented in Figure 38. In this map, the hue of each point is proportional to its estimated altitude, while saturation and intensity color coordinates are derived from the satellite imagery. The GIS annotations supply useful context.

The 3D NYC map serves as a global backdrop into which information localized in space and time may be incorporated. In order to georegister the relative SfM reconstruction with the absolute map, we select 10 photos with large angular coverage and small reconstruction uncertainties. We then manually pick 33 features in the ladar map coinciding primarily with building corners and identify counterparts to these features within the 10 photos. A least-squares fitting procedure subsequently determines the global transformation parameters needed to align all reconstructed photos with the 3D map. Figures 39 and 40 illustrate the 1000+ Flickr pictures georegistered with the NYC map.

In order to efficiently display large numbers of pictures in our OpenSceneGraph viewer, they are rendered as low-resolution thumbnails inside frusta when the virtual camera is located far away in world space. When the user clicks on some frustum, the virtual camera zooms in to look at the full-resolution version of the selected image. For example, the top row of Figure 41 illustrates a Statue of Liberty photo in front of the statue's reconstructed point

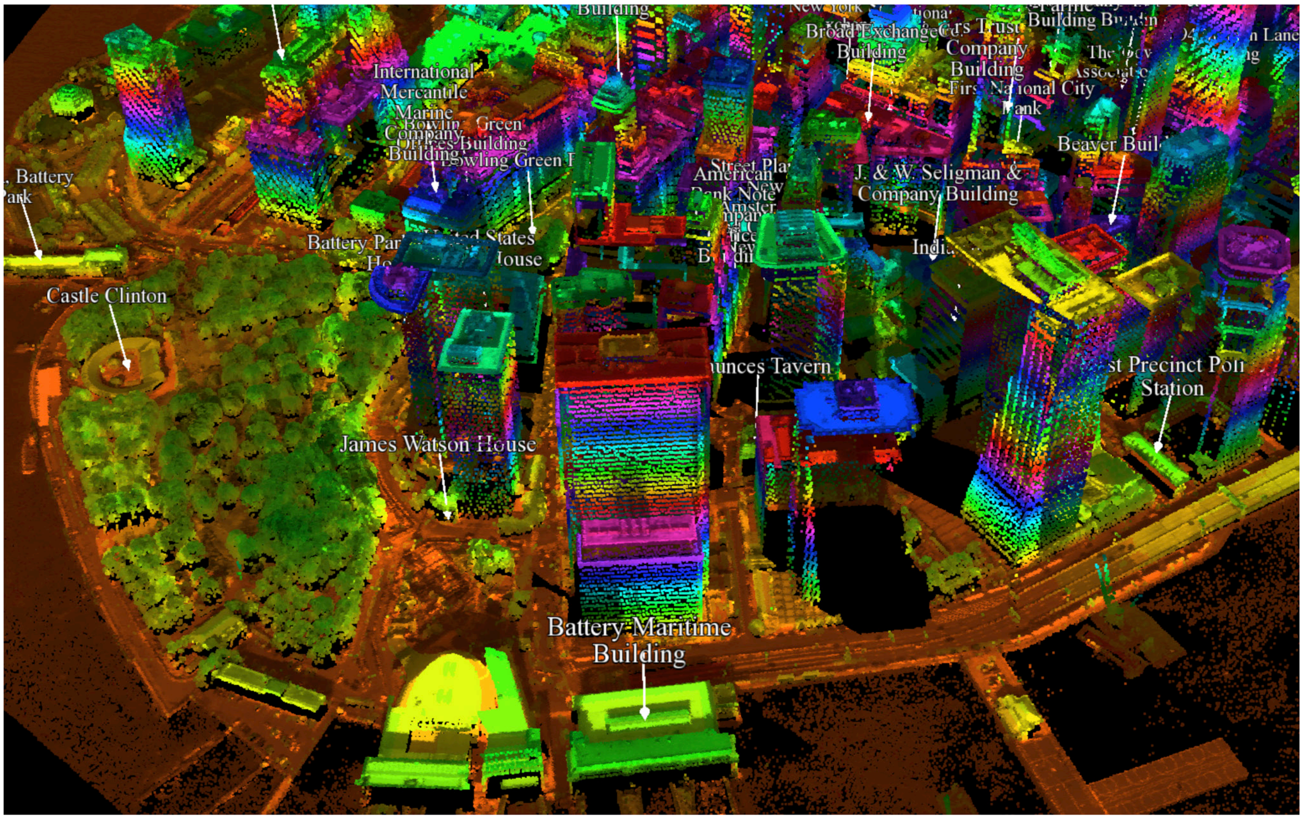


FIGURE 38. Fused 3D map of NYC. [video]

cloud (for which we do not have ladar data). By comparing geocoordinates for reconstructed points on the Statue of Liberty with their pixel counterparts in Google Earth overhead imagery, we estimate that the average angular orientation error for the georegistered Flickr cameras is approximately 0.1 degree.

A more stringent test of georegistration accuracy is provided by the alignment between projected ladar points and their corresponding image pixels, particularly for cameras located far away from their target objects. The second row in Figure 41 exhibits the match between one representative skyline photo and the ladar background. Their agreement represents a nontrivial georegistration between two completely independent datasets. Similarly good alignment holds for nearly all other skyline photos and the 3D map.

Once the reconstructed photo collection is georegistered with the NYC map, many difficult exploitation problems become tractable. Here we present four examples of geometry-based augmentation of geospatially organized pictures.

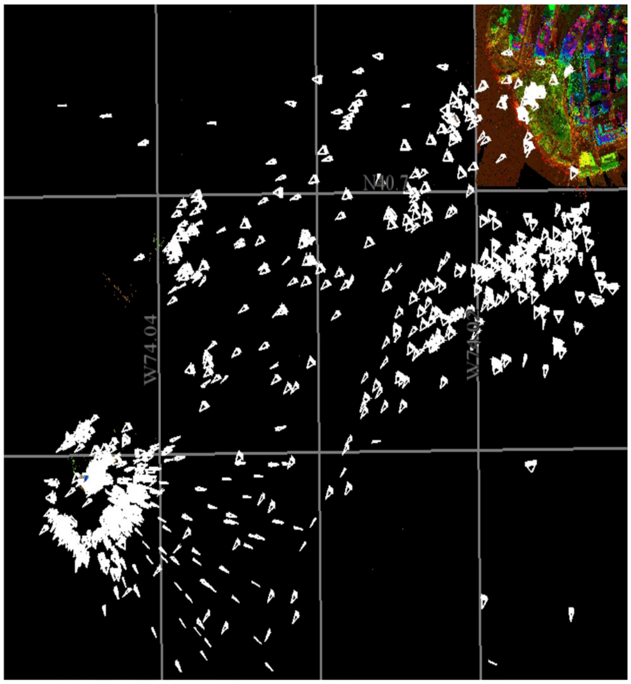
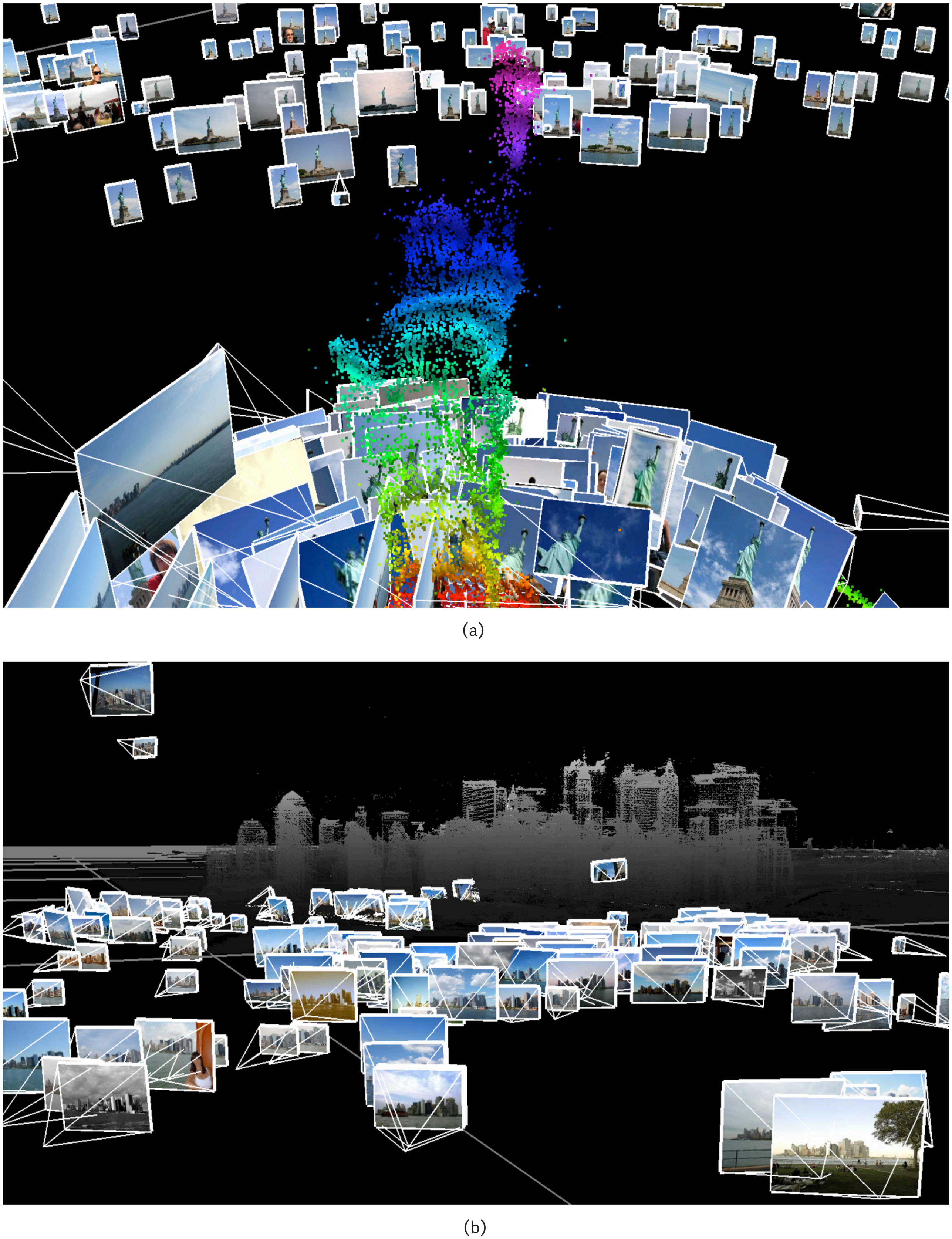


FIGURE 39. 1012 Flickr photos georegistered with the 3D NYC map. [video]



(a)

(b)

FIGURE 40. Reconstructed and georegistered Flickr photos of (a) the Statue of Liberty and (b) lower Manhattan skyline. [video]

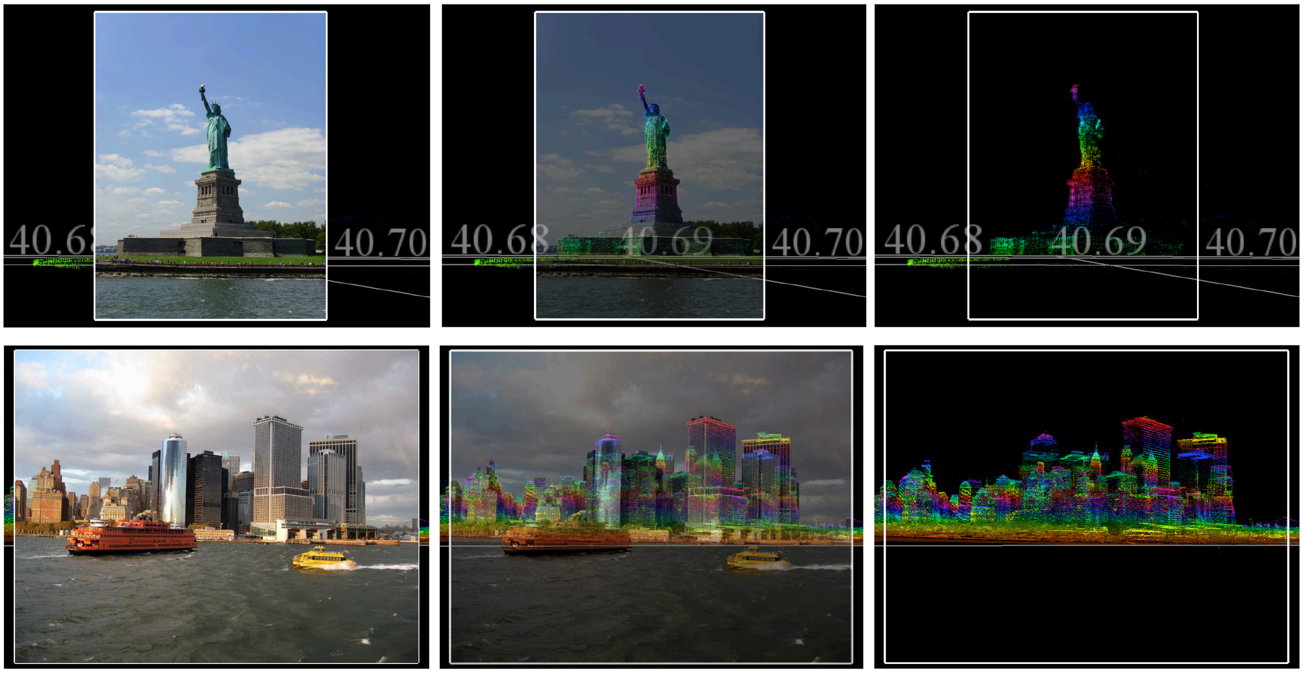


FIGURE 41. Flickr photo alignments with combined reconstructed and ladar NYC point clouds are seen after the virtual camera assumes the same position and orientation as georegistered cameras and image planes are faded away. [video]

Urban Scene Annotation

Our first example of open-source imagery exploitation is automatically annotating static objects in complex urban scenes. We specifically would like a machine to label buildings within Flickr photos of the NYC skyline. This annotation problem is extremely challenging because of the wide range of possible viewing and illumination conditions. But once a photo collection is georegistered, we leverage the fact that building names are tied to specific geolocations. After a camera has been globally reconstructed, projecting skyscraper labels into its image plane is straightforward. This basic projection approach holds for other geospatially anchored information such as roadway networks and political zones.

One technical problem for urban knowledge projection arises from line-of-sight occlusion. To overcome this issue, we convert the ladar point cloud into a height map and assume walls drop straight downward from rooftop ladar data. If a ray traced from a world-space point back to a reconstructed camera encounters a wall, the point is deemed to be occluded from the camera's view. Information associated with that point is then not used to annotate the image. We note that such raytracing works only for static occluders like buildings and not for transient occluders like people and cars. Figure 42 displays the results for annotating building names using this projection and raytracing procedure.

Image Information Transfer

Our second exploitation example demonstrates knowledge propagation between image planes. Figure 43 illustrates a prototype image-based querying tool that exhibits a Flickr photo in one window and the 1000+ georegistered frusta in another. When a user selects a pixel in the window on the left, a corresponding voxel is identified via raytracing in the map on the right. A set of 3D crosshairs marks the world-space counterpart. The geocoordinates and range for the raytraced point are returned and displayed alongside the picked pixel. Note that the ocean pixel selected in Figure 43 is reassuringly reported to lie at 0 meter above sea level.

Once a 3D point corresponding to a selected 2D pixel is identified, it may be reprojected into any other camera so long as raytracing tests for occlusion are performed. For instance, distances from different cameras to previously selected urban features are reported in Figure 44. Alternatively, static counterparts in overlapping air and ground views could be automatically matched. Future video versions of this prototype information-transfer system could even hand off tracks for dynamic

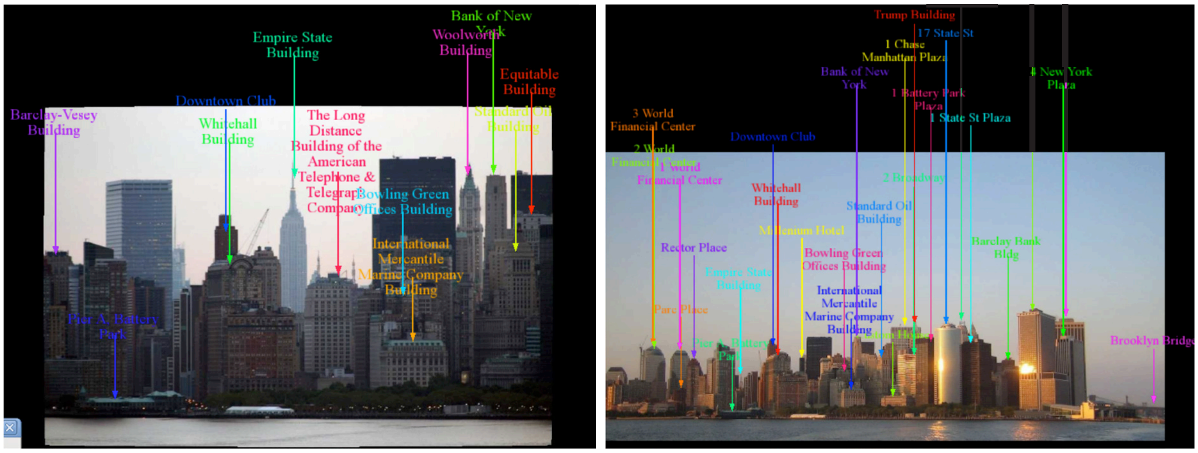
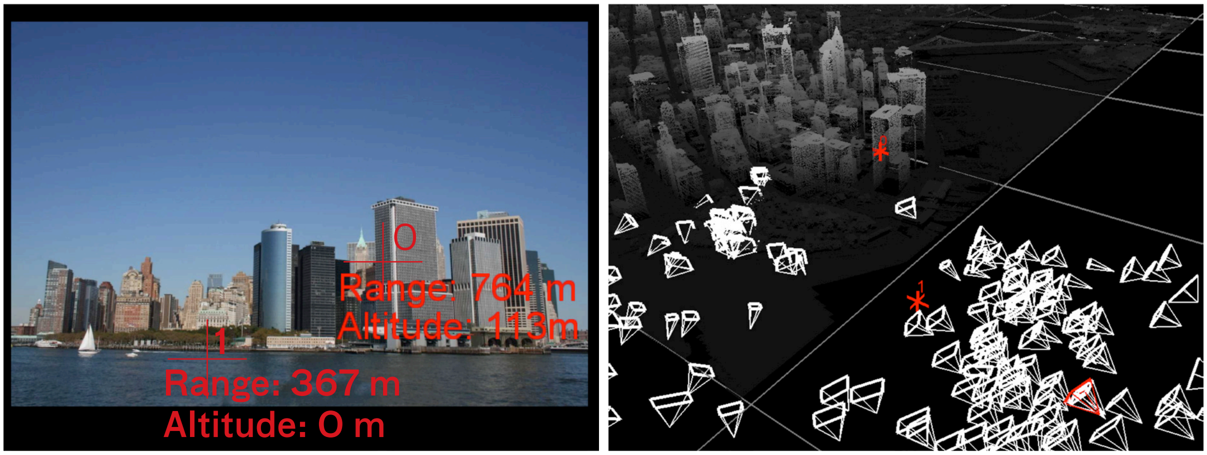


FIGURE 42. Two Flickr photos were annotated automatically by projecting building names from the 3D NYC map into their image planes.



(a) (b)

FIGURE 43. Image-based querying. (a) A user selects two pixels in a Flickr photo. The machine traces their corresponding rays back into the 3D NYC map. (b) Voxels intercepted by the rays have their ranges and altitudes displayed within the photo window. [video]

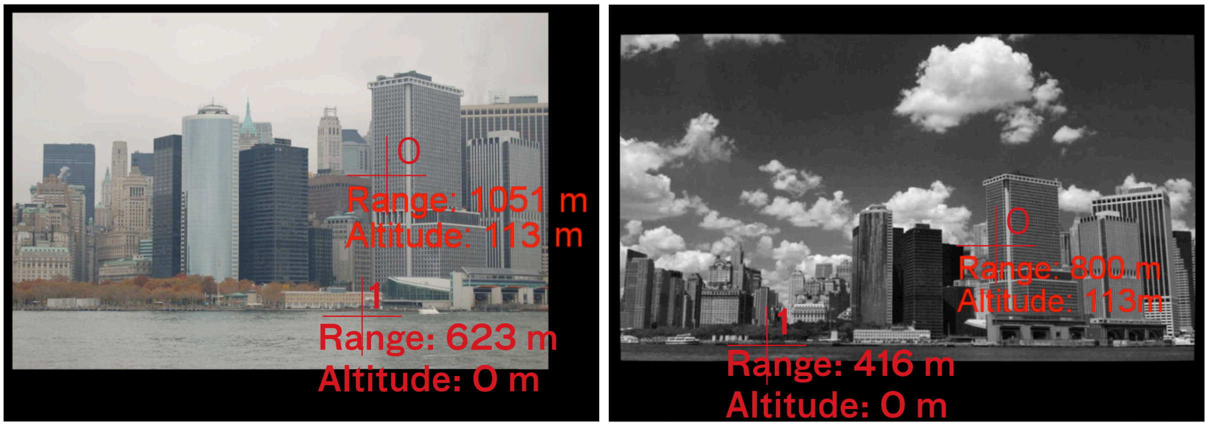


FIGURE 44. Reprojection of voxels indirectly selected in Figure 43 onto two different Flickr photos. Camera ranges to 3D voxels depend upon image, while voxel altitudes remain invariant. [video]

urban movers between multiple cameras, provided the movers' length scales are a priori known.

Image Segmentation

Image segmentation represents a classic problem in computer vision that can be dramatically simplified by geometry. For instance, suppose we want to classify every pixel in the NYC skyline photos in Figure 45a as belonging to sky, ocean, or land. Once the 2D photo is georegistered, we can backproject each of its pixels into world space. If a raytraced pixel does not intersect any point in the 3D map (with occluding walls taken into account), it is categorized as sky. Such identified sky pixels are tinted red, as shown in Figure 45b. Pixels backprojecting onto points with zero altitude above sea level are labeled as ocean and tinted blue. Finally, all pixels not classified as sky or ocean are deemed to belong to land. The resulting image segmentation is quite accurate and simple to compute. While this particular algorithm may not work in all cases (e.g., places where water is above sea level), it could be extended to handle more detailed GIS data.

Image Retrieval

Our last example of 3D exploitation is image retrieval. We present here a simple version of a gazetteer capability based upon projective geometry. Specifically, when a user enters the name of a building or landmark as a text string, our machine returns a list of photos containing that object ordered by reasonable visibility criteria.

Using the GIS layer within the 3D NYC map, the computer first looks up the geolocation for a user-specified GIS label. After performing 2D fill and symmetry decomposition operations, it fits a 3D bounding box around the ground target of interest (see centers of

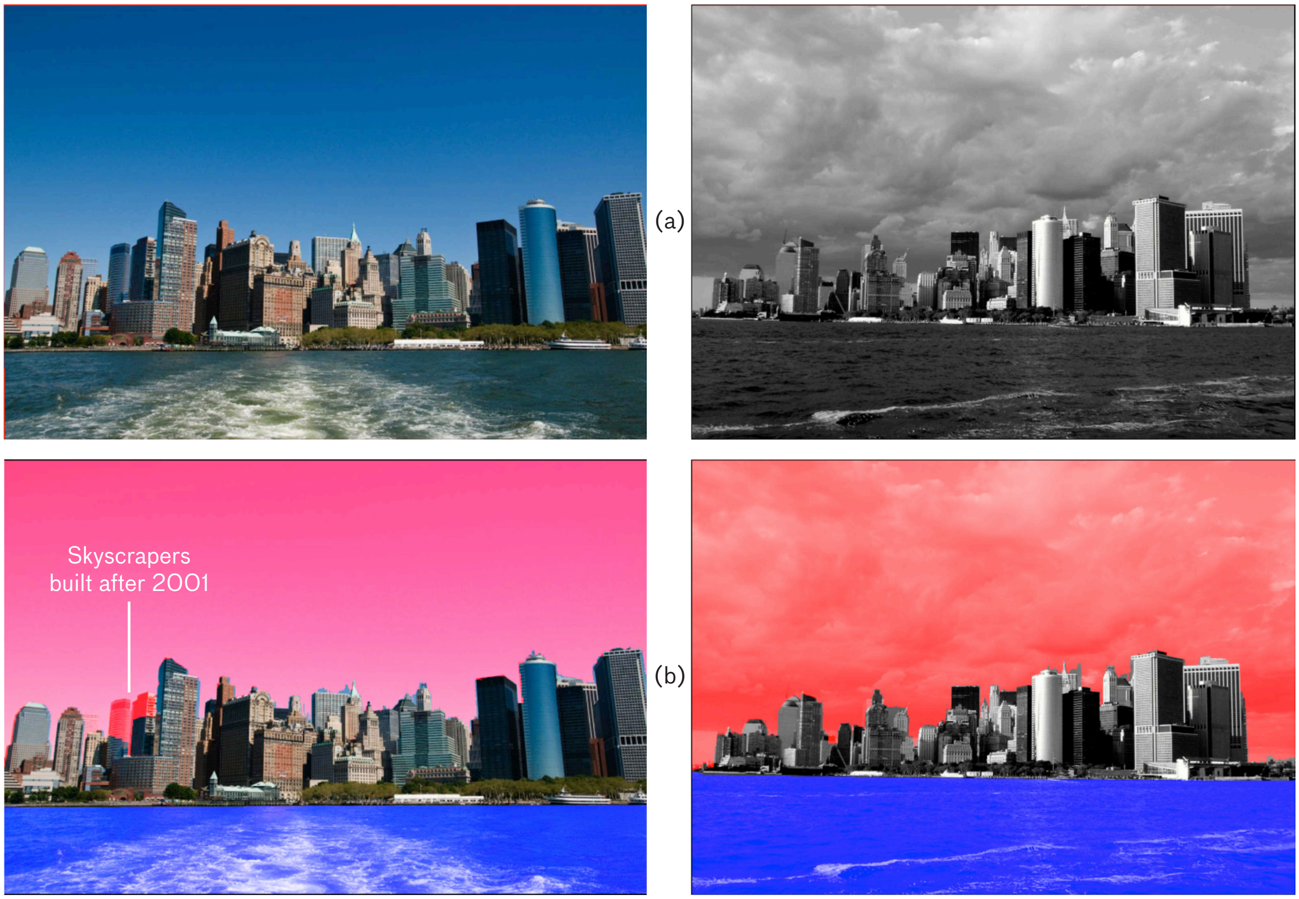


FIGURE 45. Examples of photo segmentation. (a) Flickr photos of NYC skyline. (b) Automatically classified sky [ocean] pixels are tinted red [blue]. The vertical arrow indicates locations for skyscrapers built after the Rapid Terrain Visualization (RTV) ladar data were collected in 2001.

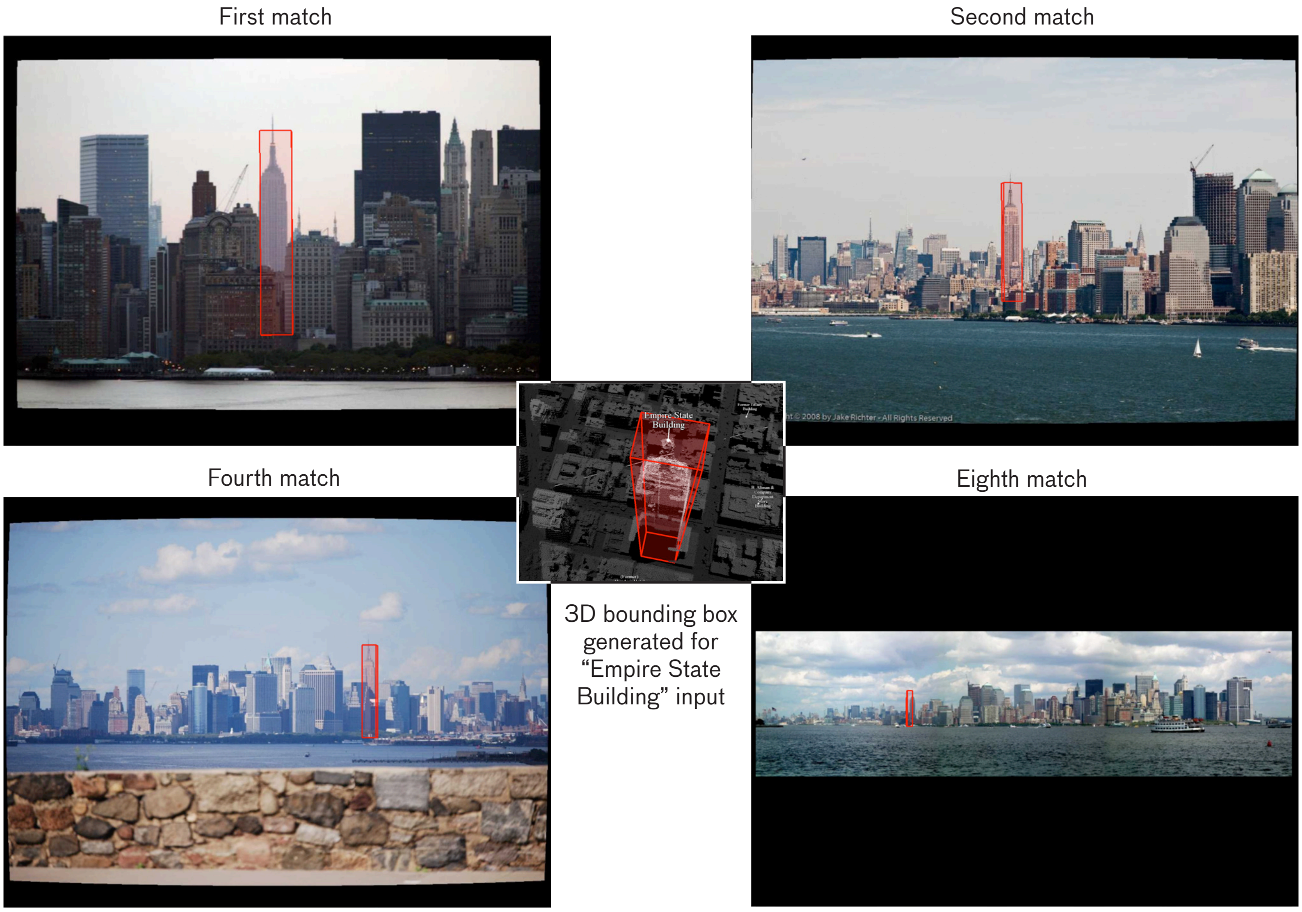

First match
Second match
Empire State Building
Fourth match
Eighth match
3D bounding box generated for "Empire State Building" input
© 2008 by Jake Richter - All Rights Reserved

FIGURE 46. Examples of image retrieval. First, second, fourth, and eighth best matches to "Empire State Building" among 1012 Flickr photos. The projection of the skyscraper's 3D bounding box is colored red in each image plane.

Figures 46 and 47). The computer subsequently projects the bounding box into each georegistered image. In some cases, the box does not intersect a reconstructed camera's field of view, or it may be completely occluded by foreground objects. But for some of the georegistered photos, the projected bounding box overlaps their pixel contents. The computer then ranks the image according to a score function comprising four multiplicative terms.

The first factor in the score function penalizes images for which the urban target is occluded. The second factor penalizes images for which the target takes up a small fractional area of the photo. The third factor penalizes zoomed-in images for which only part of the target appears inside the photo. The fourth factor weakly penalizes photos in which the target appears too far off from image plane centers. After drawing the projected bounding box within the input photos, our machine returns the annotated images sorted according to their scores.

Figure 46 illustrates the first, second, fourth, and eighth best matches to "Empire State Building" among our 1000+ Flickr photos. The computer scored relatively zoomed-in, centered, and unobstructed shots of the requested skyscraper as optimal. As one would intuitively expect, views of the building for photos located further down the sorted list become progressively more distant and cluttered. Eventually, the requested target disappears from sight altogether. We note that these are not the best possible views of the Empire State Building, as our image database covers a fairly small range of Manhattan vantage points. In contrast, the projected bounding boxes in Figure 47 corresponding to the first, second, fourth, and eighth best matches to "1 Chase Manhattan Plaza" are larger than their Empire State Building analogs. Our reconstructed skyline cameras have a better view of the downtown banking building than the iconic midtown skyscraper.

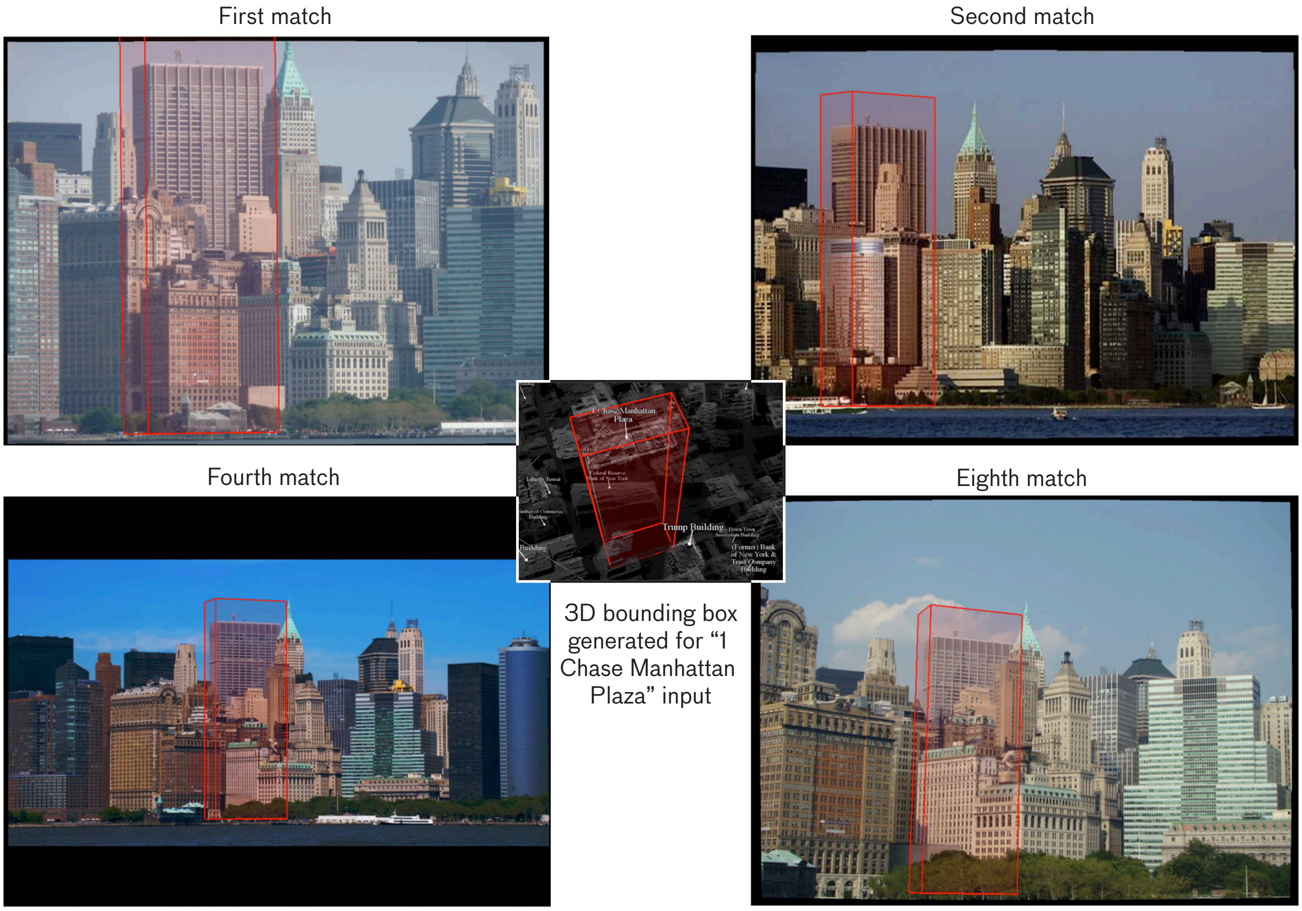

First match
Second match
1 Chase Manhattan Plaza
Federal Reserve Bank of New York
Liberty Tower
amber of Commerce Building
Building
Trump Building
Down Town Association Building
(Former) Bank of New York & Trust Company Building
Fourth match
Eighth match
3D bounding box generated for "1 Chase Manhattan Plaza" input

FIGURE 47. First, second, fourth, and eighth best matches to "1 Chase Manhattan Plaza" among 1012 Flickr photos.

Future robust versions of this image retrieval capability would provide a powerful new tool for mining imagery. Unlike current text-based search engines provided by Google, Flickr, and other web archives, our approach requires no prior human annotation of photos in order to extract static objects of interest from complex scenes. To the extent that input photos can be automatically reconstructed, the geometrical search technique is also independent of illumination conditions and temporal variations. It consequently takes advantage of the inherent geometrical organization of all images.

Ongoing and Future Work

In this article, we have demonstrated 3D exploitation of 2D imagery in multiple contexts. To appreciate the broad scope of problems that geometry can help solve, we return in Figure 48 to an extended version of the imagery exploitation applications presented in Figure 3 at the beginning of this article.

Defense and intelligence community applications are again ordered in the figure by their a priori camera uncertainty and data processing complexity. Perimeter surveillance and aerial reconnaissance involve imagery that is cooperatively collected. For such problems, partial or even complete estimates for camera parameters are often obtainable from hardware measurements. Therefore, we are currently working to exploit camera metadata accompanying cooperatively gathered imagery in order to significantly speed up its geometrical processing. Near-real-time 3D exploitation of cooperatively collected imagery will have major implications for intelligence, surveillance, and reconnaissance applications as well as robotic operations.

On the other end of the imagery-gathering spectrum lie Internet pictures that come with little or no useful camera metadata. Parallelized computer clusters are required to perform massive calculations to exploit web images originating from unorganized online archives. Prelimi-

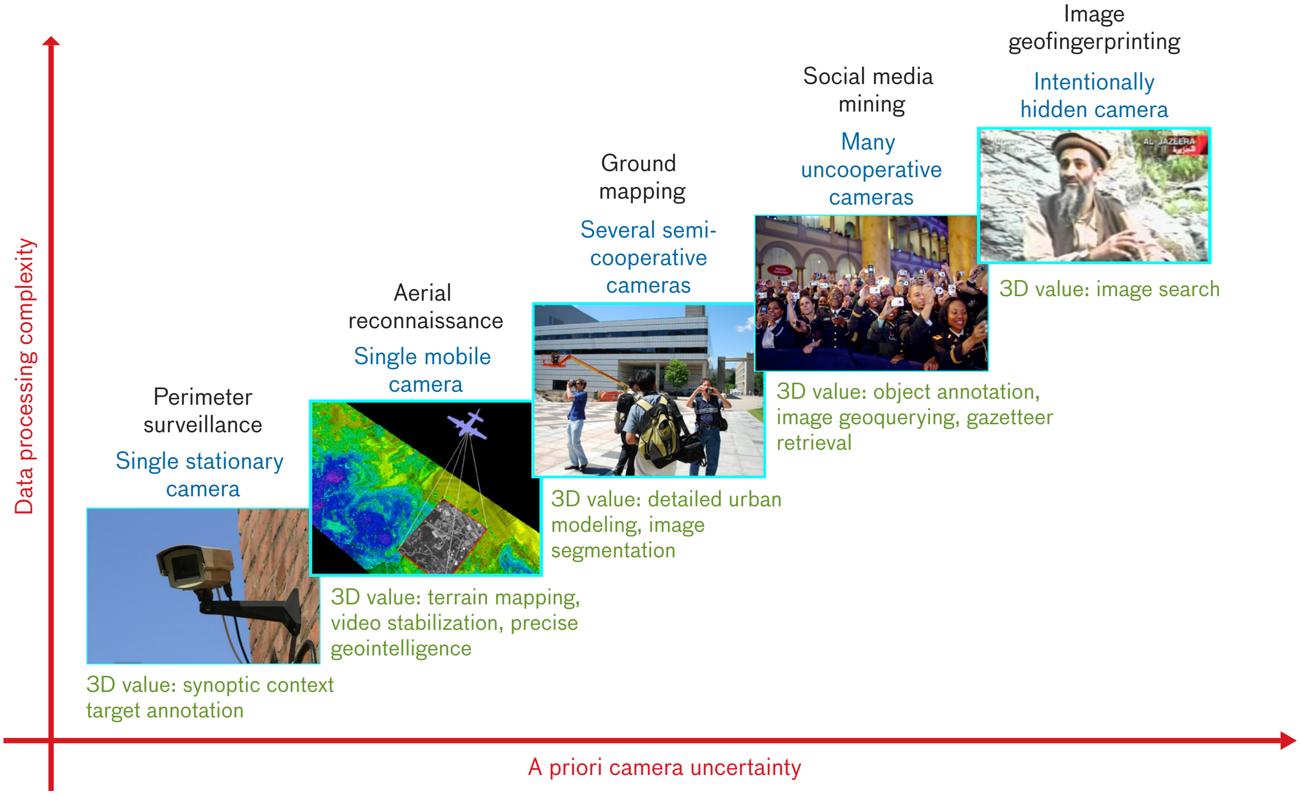


Image geofingerprinting
Intentionally hidden camera
3D value: image search
Social media mining
Many uncooperative cameras
3D value: object annotation, image geoquerying, gazetteer retrieval
Ground mapping
Several semi-cooperative cameras
3D value: detailed urban modeling, image segmentation
Aerial reconnaissance
Single mobile camera
3D value: terrain mapping, video stabilization, precise geointelligence
Perimeter surveillance
Single stationary camera
3D value: synoptic context target annotation
AL JAZEERA
Data processing complexity
A priori camera uncertainty

FIGURE 48. Department of Defense and intelligence community applications for 3D exploitation of 2D imagery.

nary experiments have demonstrated that it is sometimes possible to topologically and geometrically match pictures gleaned from the Internet with structured data repositories. When such matching is successful, intelligence can propagate from known into unknown images as we have repeatedly demonstrated throughout this article.

Looking into the future, we see computer vision inexorably moving toward assigning "fingerprints" to every digital photo and video frame. If the technical challenges associated with fingerprinting and retrieving images on an Internet scale can be overcome, it should someday be possible to search for arbitrary electronic pictures just as we search for arbitrary text strings on the web today. Much work needs to be done before arbitrary image search and geolocation will become a reality. But we look forward to continuing progress in this direction and the many interesting technical developments that will transpire along the way. ■

References

1. For a history of digital cameras, see http://en.wikipedia.org/wiki/History_of_the_camera#Digital_cameras.
2. http://www.petapixel.com/2010/08/05/the-worlds-first-digital-camera-by-kodak-and-steve-sasson/
3. M. Brown and D.G. Lowe, "Automatic Panoramic Image Stitching using Invariant Features," *International Journal of Computer Vision*, vol. 74, no. 1, 2007, pp. 59–73.
4. M. Brown, R.L. Hartley, and D. Nister, "Minimal Solutions for Panoramic Stitching," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
5. D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, 2004, pp. 91–110.
6. S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu, "An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions," *Journal of the ACM*, vol. 45, no. 6, 1998, pp. 891–923.

7 D.M. Mount, ANN Programming Manual, downloadable from http://wwww.cs.umd.edu/~mount/ANN, 2006.

8. M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, no. 6, 1981, pp. 381–395.
9. N. Snavely, S.M. Seitz, and R. Szeliski, "Photo Tourism: Exploring Photo Collections in 3D," ACM *Transactions on Graphics*, vol. 25, no. 3, 2006, pp. 835–846.
10. N. Snavely, S.M. Seitz, and R. Szeliski, "Modeling the World from Internet Photo Collections," *International Journal of Computer Vision*, vol. 80, no. 2, 2008, pp. 189–210.
11. K.S. Arun, T.S. Huang, and S.D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Transactions on Pattern*

Analysis and Machine Intelligence, vol. 9, no. 5, 1987, pp. 698–700.

12. M. Lourakis, LEVMAR, 2008, downloadable from http://www.ics.forth.gr/~lourakis/levmar/.
13. P.R. Kalata, "The Tracking Index: A Generalized Parameter for αβ and αβγ Trackers," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 20, no. 2, 1984, pp. 174–182.
14. J.E. Grays, "A Derivation of an Analytic Expression for the Tracking Index for the Alpha-Beta-Gamma Filter," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 29, no. 3, 1993, pp. 1064–1065.
15. See http://maps.yahoo.com for aerial urban imagery.
16. P. Cho, "3D Organization of 2D Urban Imagery," *Proceedings of SPIE: Signal Processing, Sensor Fusion and Target Recognition XVII,* vol. 6968, 2008.
17. P. Cho, S. Bae, and F. Durand, "Image-Based Querying of Urban Knowledge Databases," *Proceedings of SPIE: Signal Processing, Sensor Fusion and Target Recognition XVIII,* vol. 7336, 2009.
18. A. Vidan, P. Cho, M.R. Fetterman, and T. Hughes, "Distributed Robotic Systems Rapid Prototyping Challenge," AUV-SI's Unmanned Systems North America Conference, 2011.
19. N. Snavely, "Bundler: Structure from Motion (SFM) for Unordered Image Collections," 2010, http://phototour.cs.washington.edu/bundler/.
20. N. Bliss, R. Bond, J. Kepner, H. Kim, and A. Reuther, "Interactive Grid Computing at Lincoln Laboratory," *Lincoln Laboratory Journal,* vol. 16, no. 1, 2006, pp. 165–216.
21. S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, and R. Szeliski, "Building Rome in a Day," *Proceedings of the 12th International Conference on Computer Vision,* 2009, pp. 72–79.
22. Y. Furukawa and J. Ponce, "Accurate, Dense, and Robust Multiview Stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, no. 8, 2010, pp. 1362–1376.
23. P. Cho and M. Yee, "Image Search System," 22nd Applied Imagery Pattern Recognition Workshop (AIPR), 2012.
24. Office of Geographic Information (MassGIS), at the Massachusetts Executive Office for Administration and Finance, http://www.mass.gov/anf/research-and-tech/it-serv-and-support/application-serv/office-of-geographic-information-massgis/.
25. P. Cho and N. Snavely, "3D Exploitation of 2D Ground-Level and Aerial Imagery," *Proceedings of the IEEE Applied Imagery Pattern Recognition Workshop,* 2011, pp. 1–8.
26. See http://www.openscenegraph.org/projects/osg for the OpenSceneGraph 3D graphics toolkit.
27. See http://www.sketchup.com for the 3D model building SketchUp tool.
28. The Flickr photo-sharing website is accessible at http://www.flickr.com.

About the Authors

Peter Cho is member of the technical staff in the Active Optical Systems Group at Lincoln Laboratory. Since joining the Laboratory in 1998, he has conducted research spanning a broad range of subjects, including machine intelligence, computer vision, and multisensor imagery fusion. He received a bachelor's degree in physics from the California Institute of Technology (Caltech) in 1987 and a doctorate in theoretical particle physics from Harvard University in 1992. From 1992 to 1998, he worked as a postdoctoral physics research fellow at Caltech and Harvard.

Noah Snavely is an assistant professor of computer science at Cornell University, where he has been on the faculty since 2009. He works in computer graphics and computer vision, with a particular interest in using vast amounts of imagery from the Internet to reconstruct and visualize our world in 3D. He received a bachelor's degree in computer science and mathematics from the University of Arizona in 2003, and a doctorate in computer science and engineering from the University of Washington in 2008. His thesis work was the basis for Microsoft's Photosynth, a widely used tool for building 3D visualizations from photo collections. He is the recipient of a Microsoft New Faculty Fellowship and a National Science Foundation CAREER Award.