Shared Perception for Autonomous Systems

Herbert E.M. Viggh, Danelle C. Shah, Peter L. Cho, Nicholas L. Armstrong-Crews, Myra Nam, and Geoffrey Brown

Small autonomous vehicles can carry only a limited assortment of sensing and computational devices. One method of increasing these vehicles' capabilities is to access offboard, networked resources. A useful resource for robotic systems would be a three-dimensional model of the environment, continually updated and made available via netcentric applications. Researchers at Lincoln Laboratory are exploring the use of such a world model by autonomous vehicles to improve the detection of objects of interest.

Size, weight, and power constraints limit the number and types of sensors and data processors that autonomous vehicles, such as mobile robots, can carry. When an autonomous vehicle (AV) operates in a netcentric environment, it can augment its onboard capabilities by accessing resources on the network, including data collected by sensors that are on other robots, airborne platforms, and the ground, and by utilizing offboard processing resources for world modeling, mission planning, perception, and navigation. When an AV uses these offboard resources, the robotic system effectively becomes distributed, with the mobile AV platform acting as an end effector that carries only those sensors and processing capabilities needed for the AV to execute its mission. This arrangement is referred to as distributed robotics in a netcentric environment (DRONE).

The available communications bandwidth between the AV and the network will drive the overall system design of what must be on the AV and what can be on the network. The physics of the sensing and perception process, such as the required aperture, range to target, or view angle, will require that certain mission-critical sensors are carried on the mobile platform. Additionally, fast reaction for critical sensing-based decision making will mandate that other sensing and processing capabilities be integrated onto the mobile platform. However, all other processing of the sensor data collected by the AV could be sent to a network if enough bandwidth were available. Likewise, data collected by offboard sensors and information extracted from such data could be sent to the AV for processing, especially if it were combined with local sensor data.

One potentially useful resource on a network is a world model built and updated from sensors and other data sources. An example would be a three-dimensional (3D) world model built from airborne 3D lidar and video as described in Felzenszwalb et al. [1]. Such a model could be employed by an AV for many tasks: mission planning, localization, navigation, and local sensor data processing for perception of the environment.

Perception Algorithms and False-Alarm Filtering (heading level 1)

Perception algorithms for use on AVs endeavor to detect and characterize objects in the local environment. Historically, such algorithms employ only their own local sensor data and their own world models built up during their missions. However, prior knowledge embodied in a shared world model could be used to create perception algorithms that perform better or that are simplified. For example, a

perception algorithm designed to detect people in an urban environment could leverage known locations of buildings, streets, and sidewalks to reduce false detections by reasoning about the areas in which people physically can walk and thus the places within those areas in which they are more apt to walk.

The use of a shared 3D world model was investigated as a method to improve the performance of local perception algorithms. This shared perception model was built from datasets emulating collections by offboard sensor systems and other AVs. The local perception problems selected were the detection of people and cars in the urban environment of the eastern section of the Massachusetts Institute of Technology (MIT) campus, hereafter referred to as MIT East Campus.

The perception algorithm used was the deformable parts models (DPM) object detector [1]. Existing DPM software and trained DPM classifiers for detecting people and cars were used in this research [2]. The DPM algorithm takes an arbitrary image or photograph and runs a classifier across the entire two-dimensional (2D) image at various scales and generates candidate detections, with a resulting probability of detection (P_d) and probability of false alarm (P_{FA}) dependent on the final detection threshold used in the presence of noise (note that in this project, false-alarm rate per photograph was used rather than P_{FA}). Varying this detection threshold results in a typical P_{FA} versus P_d curve, commonly referred to as a receiver operating characteristic (ROC) curve. In such a classical detection problem, the detection threshold is typically calculated using the costs of events, such as missing a true detection or declaring a false detection as true (i.e., a false alarm).

The specific class of detection problems that our research addressed is one in which a high cost is associated with missing a detection. For example, a mobile robot operating in an urban war zone, where any encountered human is a potential threat, relies on precise identifications of humans. Another high cost within this type of detection problem results when resources are utilized to screen or react to false alarms. In a typical ROC curve, a high P_d is associated with a high P_{FA} because lowering the detection threshold to increase the number of true detections also tends to increase the number of false alarms. If a filtering strategy can be employed to reduce the instances of false alarms while not significantly decreasing the number of true detections, then the overall detection performance can be improved.

We employed a false-alarm filtering strategy that uses a prior 3D model of buildings and the ground to geometrically filter out detections that are physically impossible. For example, if the DPM algorithm detected a person whose size in the 2D image places him or her at a certain distance from the camera, but there is an intervening building wall between the camera and the potentially detected person, then clearly this detection is a false alarm and can be disregarded.

The above technique was implemented as a post-processing filter after the DPM algorithm was run on a photograph because existing DPM software was used and resources were not available to modify its code. Alternatively, this filtering approach could be implemented as part of the DPM processing by projecting the 3D geometry into the 2D image and then limiting the candidate object scales in different parts of the image. This second technique could reduce the overall processing load.

Different types of 3D world models were needed for two different aspects of the DPM false-alarm filtering. One critical step is finding the camera's 3D position and orientation, or pose, relative to a 3D coordinate frame. For that step, 3D point clouds of Scale Invariant Feature Transform (SIFT) features were used. The second critical step is the geometric reasoning of where detected objects can be seen, given (1) the camera's position and pose, and (2) any intervening objects and the ground. Reasonable assumptions, such as the supposition that people and cars do not float above the ground, were also applied. To ascertain objects' positions, 3D models of buildings and the ground were used. Finally,

problem-specific 3D models of areas of interest can also be used to filter for false alarms; for example, maps of known parking lots and streets can be applied to the task of detecting and counting parked cars.

Leveraged Technologies (heading level 1)

Several existing technologies supported this research effort.

Structure from Motion (heading level 2)

A significant body of work describes the structure from motion (SfM) technique for processing large sets of photographs, such as tourist photographs. The SfM approach entails detecting and matching SIFT [3] features among photographs, estimating the relative position and pose of the cameras that took them, and finally generating a 3D point cloud of the SIFT features [4]. The process for generating a camera's position and pose needed for projecting DPM detections into the 3D world can be used as the first step in false-alarm filtering. Code from Snavely et al. [4] via the Bundler website [5] and from the VisualSfM [6] software package was used on this project.

Localization Technique (heading level 2)

The SfM approach processes all N photographs together to estimate position and pose of all the camera instances that took them. This step can be thought of as a mobile robotics post-processing task in which an AV collects a set of photographs and then builds a 3D model of the SIFT features in the scene. A related problem is to take an $N + 1$st photograph and match it to a preexisting 3D point cloud of SIFT features. A robot could use such a capability in the DRONE context to estimate its position and pose by taking a photograph and matching it to a prior world model of SIFT features generated by other robots, other sensors, or tourist photographs.

Two previous Lincoln Laboratory programs explored this $N + 1$st photograph approach to estimating pose and position. The technique was developed under the Scalable Image Graph Matching and Analysis (SIGMA) program [7, 8] and employed in a previous DRONE program [9] that explored its use in robot navigation.

Deformable Part Models (heading level 2)

Object detection, which is one of the fundamental problems in computer vision and robotics, can help a navigating AV localize a target or obstacles. However, this detection is challenging because an object's appearance varies greatly because of illumination changes, various poses, and nonrigid deformation of the object's parts.

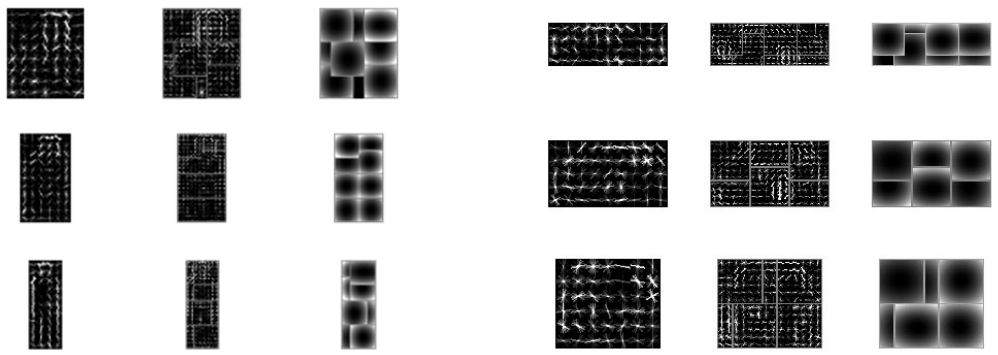
Deformable part models (DPM) detect a generic object by localizing its parts in a deformable configuration [1]. DPMs learn visual grammars parameterized by the appearance of each part and a geometric model that captures spatial relationships among parts. The DPM system has gained popularity for discrimination tasks, and it has achieved state-of-the-art results in international PASCAL Visual Object Classes Recognition challenges [10].

DPMs use supervised learning to train linear filters that detect deformable parts of objects. The filters are templates that represent histogram-oriented gradient features. Templates are built in a pyramid structure so that a coarse global template covers an entire object and higher-resolution parts templates cover individual parts of the object that may move relative to each other, such as an arm or a leg. Parts templates are computed at multiple resolution levels, and these hierarchical templates form a feature pyramid. Scores that represent the confidence that the filters have detected a part are computed from the filter responses at different locations in the image. A final DPM score for the object model is determined

by combining these individual part scores with variable constraints that limit the allowable relative position and orientation of parts with respect to each other. A detection threshold is then applied to the final score to decide whether an object is detected or not.

The final outputs for each detection are the upper-right and lower-left vertices of a rectangular bounding box with a final confidence score. From the vertices, the pixel location of the bounding box in the 2D image is known, and the area of the bounding box can be calculated. Note that detections of objects closer to the camera will have larger bounding boxes.

We obtained DPMs for human and car detections from the Discriminatively Trained DPMs Release Version 4 [2]. Figure 1 illustrates human and car templates trained with the Visual Object Classes 2009 Challenge dataset. Each object model has three components. For example, the human model is categorized into body part from head to chest, body part from head to waist, and full body. The car model has three components that specify car appearances from different viewing perspectives, such as side, front, or backend views. Each component has a root filter and eight part filters. For object detection, left-right flipped versions of the three components are added per class, so each object model ends up with six components. Figure 2 illustrates detected bounding boxes for human and car detection.



(a) (b)

FIGURE 1. These examples show a human deformable part model (DPM) (a) and a car DPM (b). For each DPM, the first column indicates the root filter, the second column is part filters, and the third column is the deformation model. Each row indicates the component.

(a) (b)

FIGURE 2. Bounding boxes are shown for a human detection (a) and a car detection (b). We applied detection threshold values to the DPM scores recommended by the DPM developers [1]: –0.5288 for human detection and –1.2009 for car detection. For each candidate object detection whose combined score is larger than the detection threshold, DPM outputs a bounding box with deformed part locations.

Exploiting the Photograph Dataset (heading level 1)

A 2009 set of photographs used in earlier work on developing 3D models from 2D images and several existing technologies were employed to support this research effort.

Photograph Dataset (heading level 2)

We built on the work of Cho and Snavely [11], who used a set of 2317 tourist-like photographs of MIT East Campus from July 2009 and SfM techniques to build a georegistered 3D model of SIFT features in that urban area. A by-product of this work was the geolocation and pose of the photographs used. We obtained this dataset to use in emulating photographs taken by a robot whose 3D position and camera pose are known.

Bias Removal (heading level 2)

The dataset of 2317 MIT photographs (henceforth referred to simply as MIT2317) represents a typical visual environment that people encounter in daily urban life. However, MIT2317 is biased because it is an aggregate of photographs taken by multiple people and contains repeating scenes that feature the same objects. Including detection results from multiple identical or similar scenes would bias the ROC-curve performance. Therefore, we removed the bias resulting from scene and object redundancy in MIT2317 to make the dataset comparable to the entire population of the unknown real visual world scenes. This bias removal is depicted in Figure 3. To remove bias, we selected a subset of images from MIT2317 that maximized the variation in background, object numbers, and their relative distances to the camera sensor by clustering the dataset and selecting the images representing individual clusters. Our algorithm performance on this selected subset would then be extensible to other unbiased datasets.

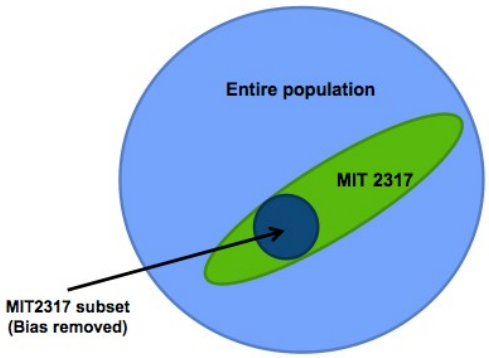


FIGURE 3. In this visual representation of bias-removal in the MIT2317 photograph dataset, the chosen subset removes multiple redundant images of the same scene and is more representative of the scene variation that would be encountered by a single robotic platform.

Bias Removal Algorithm (heading level 2)

To address background variation, we first performed scene clustering so that each cluster contained the same scene. We clustered scenes in MIT2137 using spectral clustering. Spectral clustering is useful when images might not form convex clusters in the non-Euclidean image space [12]. Given n image data points in image set S, $S = \{s_1, s_2, \dots, s_n\}$, $(n = 2317)$, we clustered the set S into K subsets. We first computed an $n \times n$ matrix A, whose element $A_{i,j}$ represents the perceptual scene distance between two images i and j. We computed a Laplacian $L = D^{1/2}AD^{-1/2}$, where D = diag(row sums of A). We found K largest (column) eigenvectors of D and arranged them in an $n \times k$ matrix, X. Then, we performed k-means clustering to obtain K clusters.

The matrix A was formed using histogram intersection as a scene-distance metric that measures perceptual scene similarities [13, 14]. Scene matching was performed by the spatial pyramid histogram matching on SIFT features extracted from individual photos [13]. The spatial pyramid matching code used can be found at a website hosted at the Department of Computer Science at the University of Illinois at Urbana-Champaign [15]. The multiscale histogram intersection metric H was learned from MIT2317, specifying image similarities. The $n \times n$ matrix A is defined as $A_{i,j} = H(s_i, s_j), i \neq j$ and $A_{i,i} = 0, i, j = 1, \dots, n$.

We chose $K = 50$ such that the clusters had optimally minimal within-cluster variation and thereby obtained a minimal image subset with maximum scene variation. While this clustering step maximized the variation among scenes, across the scenes each cluster could have inconsistent detection performance caused by illumination variation, slight camera pose changes, different object types, or the objects' changed locations. We divided each scene cluster into multiple subclusters in terms of DPM detection numbers and bounding box sizes, which could specify such changes in the same scenes.

Subgrouping with maximum variation in detection numbers was performed by computing a histogram of differences in detection numbers at two different DPM thresholds. From eight thresholds, we chose two consecutive DPM thresholds (–0.8 and –0.4) that significantly decrease the positive detections. The detection number differences at the two DPM thresholds can effectively address various scenarios with false-alarm reduction because detections at threshold –0.8 are grossly true positives while those at threshold –0.4 include both true and false positives. A sparse histogram was computed with fixed bin size covering the wide range of differences in detection number. We randomly selected one image per histogram bin and none if the bin was empty for all clusters. To obtain a subgroup with maximum

variation in detection sizes, we repeated the same process using the total area of DPM detection bounding boxes.

Using the method described above, we created an unbiased image subset of MIT2317 for testing the performance of both the baseline DPM algorithm and our new algorithm for DPM plus false-alarm filtering. The resulting MIT2317 subset contained 495 images for people detection and 464 images for car detection.

Ground Truthing (heading level 2)

To measure the performance of a detection algorithm on the objects of interest in a set of photographs, one requires the "ground truth" knowledge of the objects that are actually in the photographs and their locations within the photographs. This ground truth can then be used to measure the percentage of existing objects detected and the number of false detections (false alarms) generated; subsequently, this information can be applied to estimating the probability of detection (P_d) and probability of false alarm (P_{FA}) for a given detection threshold. Calculated at multiple thresholds, these results are used to plot the ROC curve.

Ground truthing must usually be done manually. We developed a MATLAB®-based graphical user interface (GUI) for research team members to view the MIT2317 subset and manually define rectangular bounding boxes around people and cars. The users were also asked to estimate what percentage of the height and width of the person or car was visible in the photograph. Height ratio is defined as the ratio of the height of a visible part to the full height of an object. Width ratio is the ratio of the width of a visible part to the full width of an object. Height ratio, especially, is a key factor in our performance evaluation because it determines the detection rate of DPM false-alarm filtering. In addition, the GUI assisted users in navigating through the test datasets and displayed previously detected bounding boxes.

The GUI software recorded the truth bounding box locations, box dimensions, and visible height and width ratios estimated by the user. The output of the ground truthing GUI was saved as an $n \times 4$ matrix, where n is the number of bounding boxes per image. Each row in the matrix indicates the pixel coordinates of the box, $[r_1\ c_1\ r_2\ c_2]$, where $[r_1\ c_1]$ is the row and column coordinates of the upper left vertex of the bounding box and $[r_2\ c_2]$ is the row and column of the lower right vertex of the bounding box ($r_1 < r_2$, $c_1 < c_2$). Objects that were at least 20% to 30% visible in both height and width were recorded with the estimated height/width ratios. Objects that had low height/width ratios were then discarded, and ones that were 70% or more visible were taken as ground truth and used in our detection performance analysis.

Manual Ground Truthing Instructions for the Subset of MIT2317 Photographs (heading level 3)

Ground truthing requires somewhat subjective decisions. Human errors are unavoidable because it is impossible to draw a bounding box that specifies a full extent of an object with heavy occlusion or a tilted surface. Because object appearance can be truncated, the bounding box may not correspond to the full area of the object, e.g., an image of a person from the chest up. Another issue is occlusion by another object as illustrated in Figure 4. In addition, some images contain objects that are visible but unidentifiable without the use of context. Four different team members ground truthed the dataset. We provided written instructions to each team member to keep the ground truth as consistent and objective as possible.

FIGURE 4. The DPM results show bounding boxes around cars occluded by other objects, e.g., person, pole, fence, or bushes.

MIT2317 SUBSET PEOPLE DETECTIONS (HEADING LEVEL 4)

The ground truthing instructions used for people detections are as follows:

1. Draw a bounding box on the visible part of a person as "tight" as possible.
2. Allow for variations in a person's size or appearance resulting from different viewpoints.
3. Draw a bounding box as far as a person's head and neckline if these parts are not occluded. Draw a bounding box when other body parts, e.g., leg, arm, and torso, are occluded.
4. Draw a bounding box specifying the visible extent of a human, even when a person is behind a fence.

MIT2317 SUBSET CAR DETECTIONS (HEADING LEVEL 4)

The ground truthing instructions used for car detections are as follows:

1. Draw a bounding box on a visible portion of the car only.
2. A car appearance can be divided into multiple segments by an obstacle. This segmentation happens when a car is occluded by a pole, person, bush, fence, or other object. Draw a bounding box on the entire visible portion of the car.
3. The height ratio is defined as {height of the car that appears in the image}/{actual height of the car if there is no occlusion}. The numerator is the fact, and the denominator is an estimate made by using your perception. The same rule applies to the width ratio: {width of the car that appears in the image}/{actual width of the car if there is no occlusion}. For the width ratio, ignore the car orientation.
4. DPM detects pickup trucks, minivans, and utility vans. It detects just the front parts of utility vans because of their sedan shape. Mark pickup trucks and vans as well as sedans.

Three-Dimensional (3D) Model Construction (heading level 1)

Building Models (heading level 2)

Three-dimensional models of the buildings (and ground) of MIT East Campus were needed to implement geometric false-alarm filtering. Our approach to generating 3D models of the MIT East Campus buildings involved manually finding footprint outlines of building roofs in aerial photos and then using airborne lidar to define the height of the buildings.

To pursue this approach, orthorectified aerial imagery collected over MIT East Campus was obtained from the Massachusetts Office of Geographic Information (MassGIS) website [16]. The airborne lidar was used to determine the building heights. Representative samples of the 2D and 3D imagery over MIT East Campus are displayed in Figure 5.

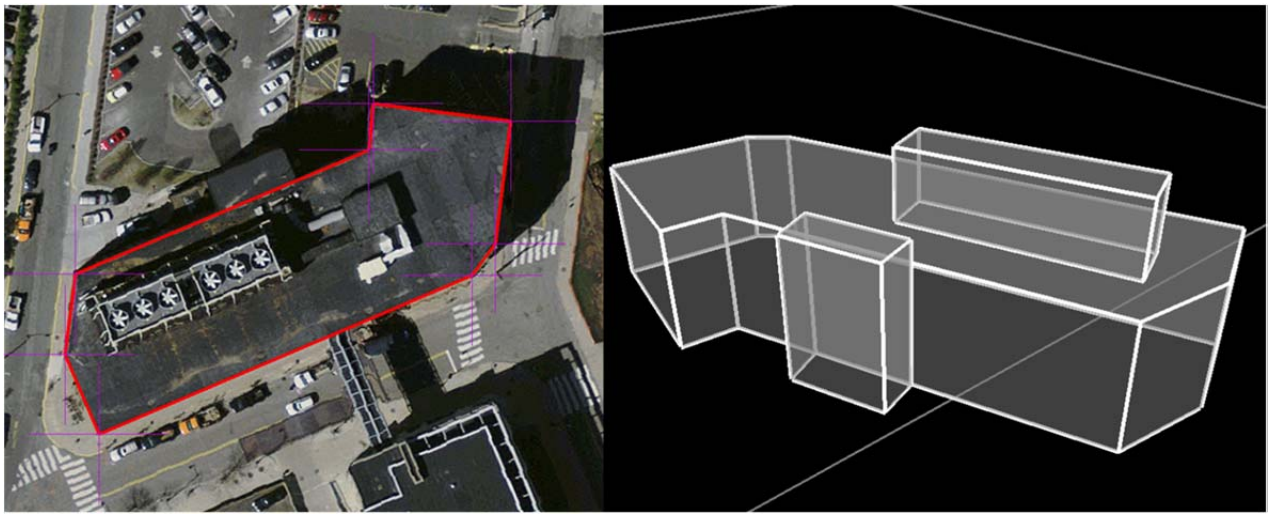
Photographs and ladar point clouds represent low-level data products that contain millions of pixels and voxels. For data-fusion purposes, it is much more useful to work with higher-level models that abstract out geometrical invariants common to all views. Consequently, a semiautomatic method was developed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

First, footprints were manually extracted from the orthorectified aerial photographs (Figure 6a). Each footprint corresponded to some part of a building with approximately constant height. Judgment was exercised as to a reasonable contour level of detail for complex urban structures. After we extracted 2D footprints of buildings, we developed software to automatically extrude them in the z direction using lidar data to determine absolute heights above sea level. The resulting prisms capture basic shape information for a single building (Figure 6b).

This semiautomatic modeling procedure was applied to 29 buildings around MIT East Campus. The models appear superposed against the lidar point cloud fused with the orthorectified aerial image in Figure 7. It is worth noting that the ground surface for this part of eastern Cambridge, Massachusetts, is well represented by a constant $z = 2.5$-meter plane (relative to sea level). This ground plane may also be simply modeled via a 3D prism, but it is not displayed in the figure for clarity's sake.

(a) (b)

FIGURE 5. These images are representative samples of 2D (a) and 3D (b) aerial imagery of MIT East Campus.



(a) (b)

FIGURE 6. A single building footprint (a) is extracted from an aerial image; after the image is processed using lidar data, the 3D building model (b) is generated.

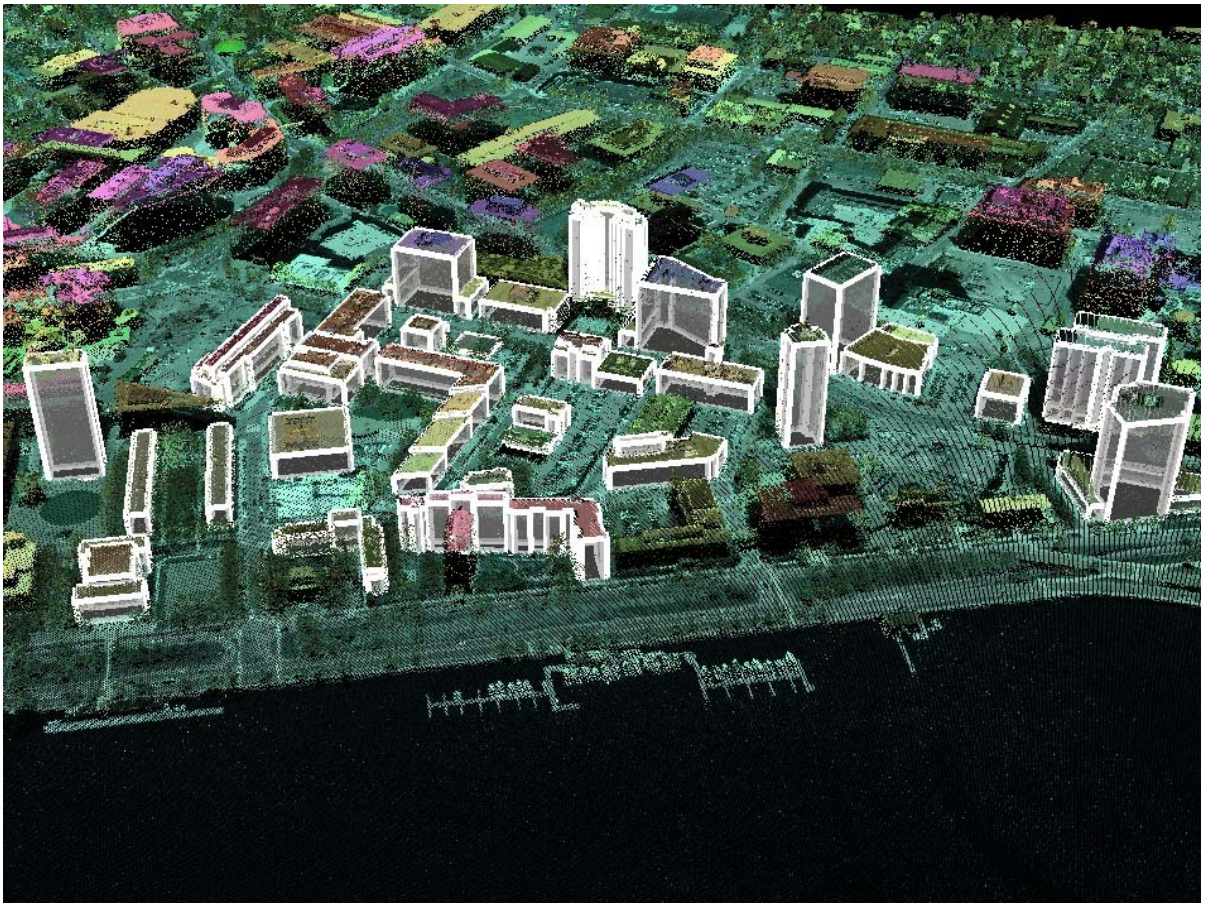


FIGURE 7. Building models are superposed against the lidar point cloud fused with the orthorectified aerial image.

Parking Lot and Road Models (heading level 2)

In addition to exploiting 3D models of buildings and the ground for filtering DPM false alarms on the basis of line-of-sight constraints, we also applied a priori knowledge of land use in an urban area. For example, if the mission is to detect parked cars, then any candidate detections within areas in which cars do not park can be assumed to be false alarms. Therefore, we developed additional 3D models describing areas in which cars are likely to park—primarily, roads and parking lots.

To create a 3D model of a parking lot, a set of points was manually chosen within the image in Figure 8 to form the outer boundary of the parking lot from the aerial viewpoint; then, the image georegistration was used to obtain the Universal Transverse Mercator (UTM) coordinates of the parking lot boundary points. Using alpha shapes [17] and MATLAB®'s inpolygon function, we connected these points with a collection of ground triangles and vertically extruded the triangles to fully represent the parking lot as a 3D volume with a predetermined height. We chose a height of 2 meters because the center of a car is generally not higher off the ground than this; therefore, cars in the parking lot should fall within the interior of our 3D parking lot model. The same procedure was used to generate the road models. Figure 9 shows a close-up of a parking lot model in MeshLab. Figure 10 shows the entire collection of road and parking lot models within MIT East Campus. Figure 11 shows the road and parking lot models combined with the building models described previously.

FIGURE 8. The georegistered image is used to manually choose points that outline the parking lots and roads.

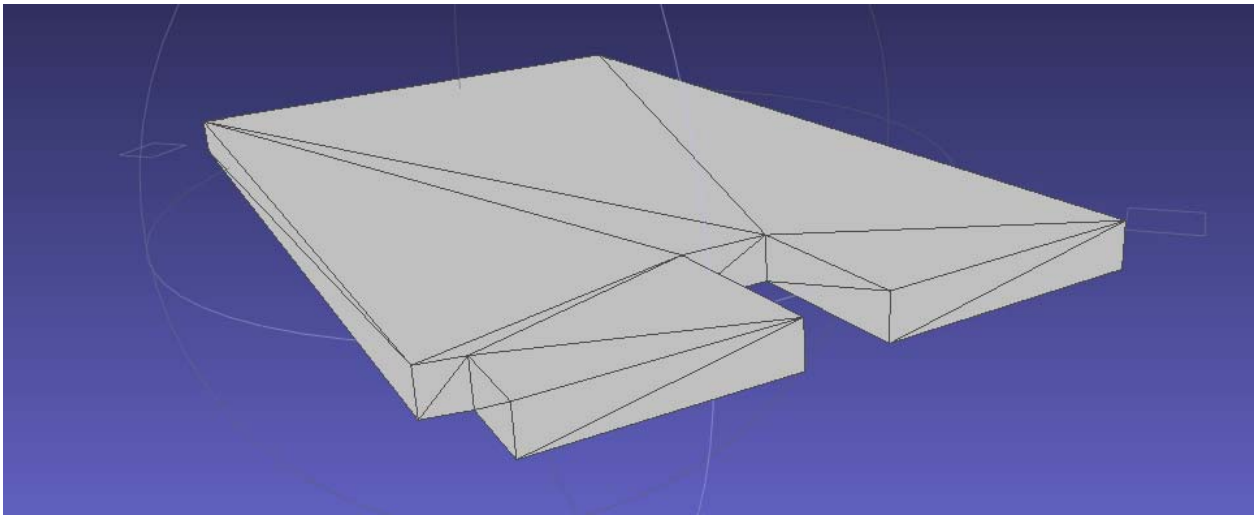


FIGURE 9. The completed 3D MeshLab model depicts a parking lot .

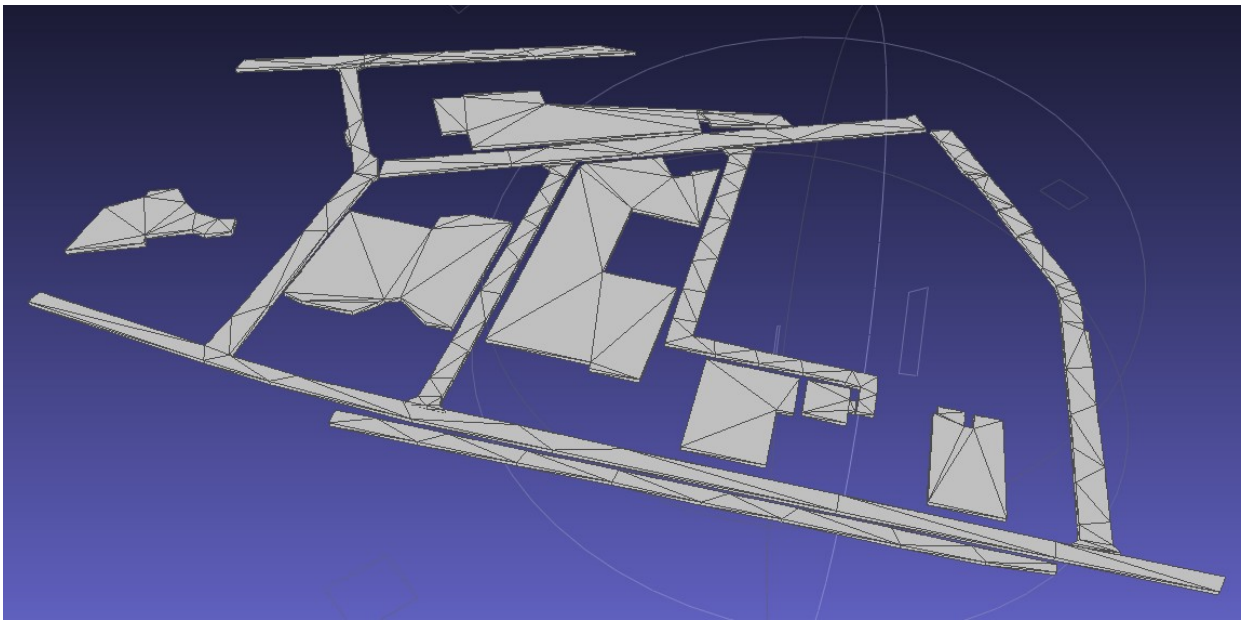


FIGURE 10. The complete 3D model shows all the roads and parking lots on MIT East Campus.

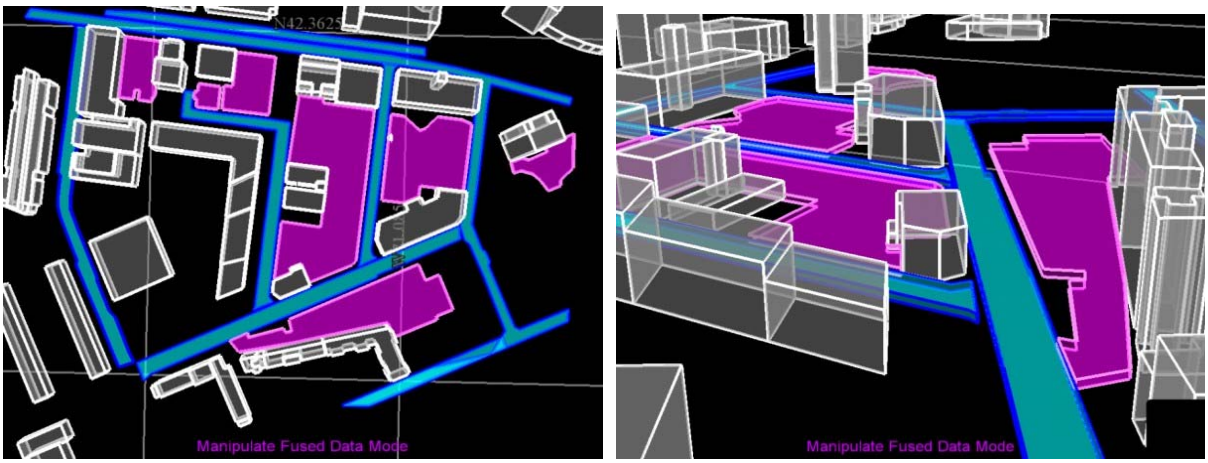


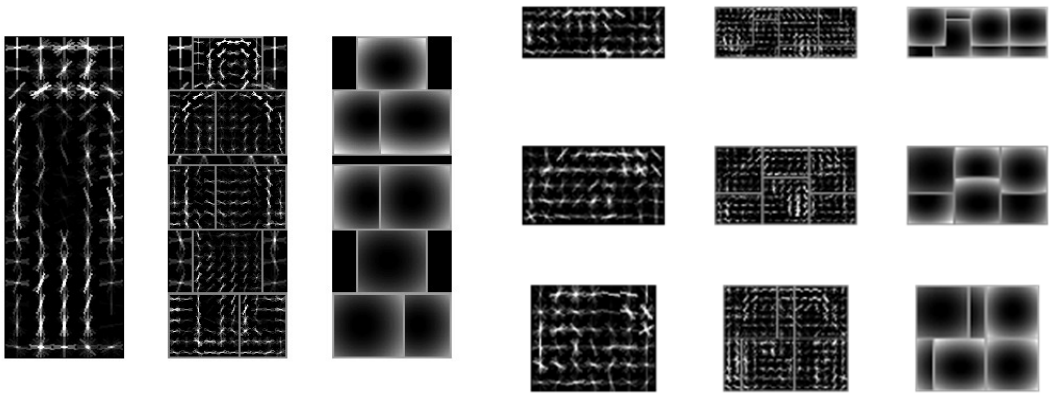
FIGURE 11. Parking lot (magenta) and road (blue and teal) models are integrated with the 3D building models (gray) to create a 3D visualization of an MIT East Campus location.

DPM Baseline Performance (heading level 1)

DPM Processing of MIT2317 Subset (heading level 2)

We ran the DPM algorithm on the MIT2317 subset to establish a performance baseline to which we could compare the false-alarm filtering results. The DPM software can be configured to only report detections above a specified confidence threshold. To build our baseline ROC performance curve, we wanted results for eight thresholds. However, rerunning the software eight times was computationally expensive. Instead, we ran it with a single very low detection threshold and then thresholded on the confidence scores reported for each detection. The eight confidence score thresholds used were [–2, –1.6, –1.2, –0.8, –0.4, 0, 0.4, and 0.8]. This threshold set will be referred to as t_{DPM}.

In this project, we used the human and car models trained by DPM Release Version 4 [2] as illustrated in Figure 12.



(a) (b)

FIGURE 12. The models used in the MIT2317 detections include the human model (a) and car model (b). The human model has a single component of an entire human body. The car model includes three different components.

Batch Processing and Resulting Datasets (heading level 2)

Lincoln Laboratory's supercomputing parallel computing cluster, LLGrid, was used to run the DPM software on the MIT2317 subset. DPM software outputs detected bounding boxes in pixel locations, with corresponding total confidence scores and parts' locations. In this project, we modified the DPM output to be an $n \times 5$ matrix indicating the exact location of a detected bounding box and its total confidence score. The row n is the number of detections per image, and each column includes the following content:

$[c_1\ r_1\ c_2\ r_2$ confidence$]$

$[c_1\ r_1]$: column and row of upper left vertex

$[c_2\ r_2]$: column and row of lower right vertex.

Figures 13 and 14 show example DPM results for the eight different confidence-score detection thresholds on human and car detection, respectively.

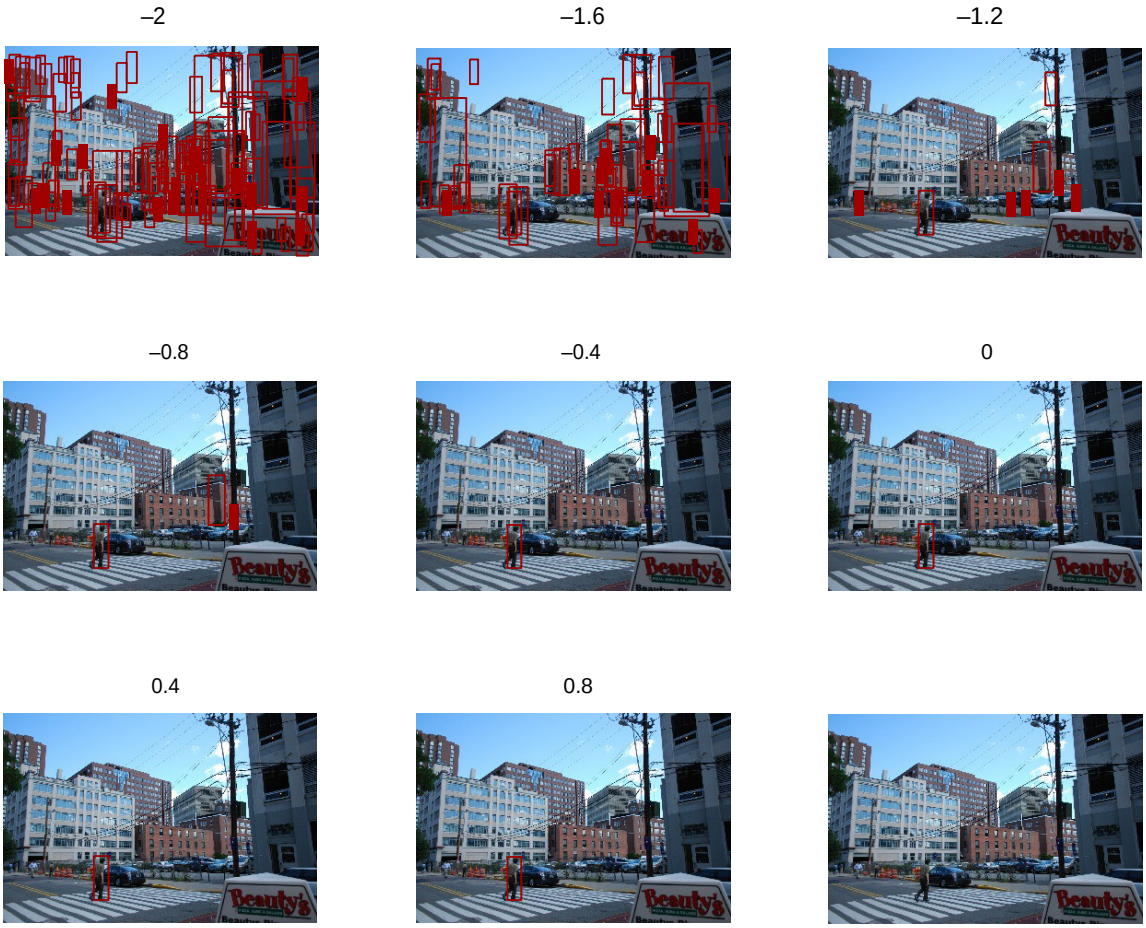


FIGURE 13. The DPM results for people detection on the MIT2317 dataset are shown for eight different thresholds for the photograph in the lower right-hand corner.

FIGURE 14. The DPM results for car detection on the MIT2317 dataset are shown for eight different thresholds for the photograph in the lower right-hand corner.

Analysis and Results (heading level 2)

DPM results were evaluated according to the PASCAL Challenge Evaluation [18]. Each detection is considered a true detection or a false positive on the basis of the area of overlap with ground-truth bounding boxes. A true detection must exceed 50% by the formula A_0,

$$A_0 = \frac{\text{area}(B_P \cap B_{GT})}{\text{area}(B_P \cup B_{GT})},$$

where B_P is the predicted bounding box and B_{GT} is the ground-truth bounding box.

In the case of multiple detections of the same object, only the one with the highest detection score is the true positive. The remaining detections of the same object are false positives. For each of the eight different thresholds in the human and car detections, we computed the four classes specified in Table 1.

TABLE 1. Classification Tasks

	Classified Positives	Classified Negatives
True examples	True positives (TP, hit)	False negatives (FN, miss)
False examples	False positives (FP, false alarm)	True negatives (TN, correct rejection)

We evaluated baseline DPM performance with three different approaches. First, we computed the four classes using bounding box area. Second, we computed them using the number of detections. In both cases, at the *i*th threshold ($i = 2, \ldots, 8$), the true negative (TN) is defined as the difference between the false positive (FP) at the current threshold and FP at the threshold smaller than the current threshold, $|FP_i - FP_{i-1}|$. Then, we computed ROC curves by computing the true positive rate TP/(TP + FN) and false positive rate FA/(FA + TN). Third, in an approach known as precision and recall (hereafter, precision-recall), three values are computed: Recall is defined as TP/(TP + FN), precision as TP/(FP + TP), and average precision (AP) as the area under the precision-recall curve.

In classification problems, the four classes are well defined. Although a ROC curve is a customary choice for classification performance evaluation, in object-detection problems, precision-recall curves are commonly used because TN is not well defined. In a ROC curve, the upper left corner, where true positives are high and false positives are low, guarantees good performance. On the other hand, in a precision-recall curve, the upper right corner with large AP indicates good detection performance.

Figures 15 and 16 illustrate DPM performance in the MIT2317 subset for human and car detections respectively. The results show that the area-based ROC curve tends to have higher true positive rates than does the ROC curve illustrating the number of detections. Human detection performs better than car detection performs in the given dataset.

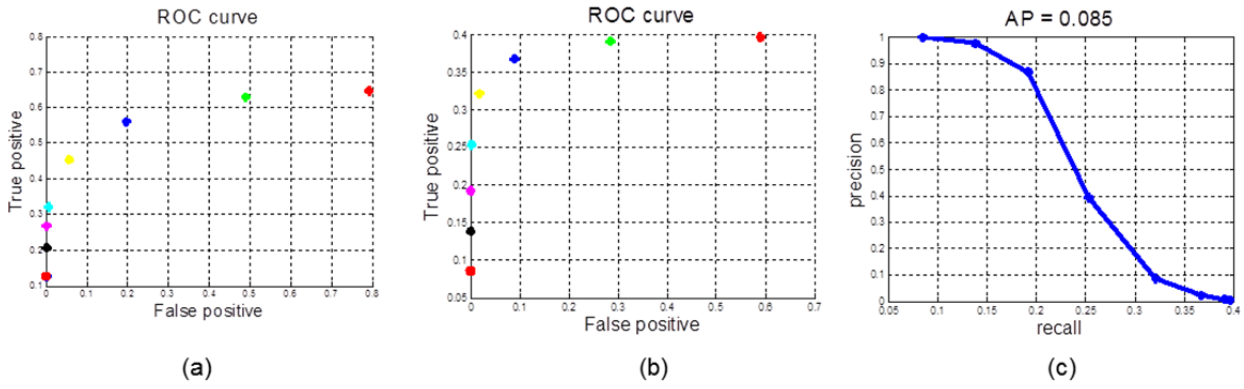


FIGURE 15. Results are shown for three approaches to human detection in the MIT2317 photographs: ROC curve for the approach using the area (a), ROC curve using the number of detections (b), and precision-recall curve (c).

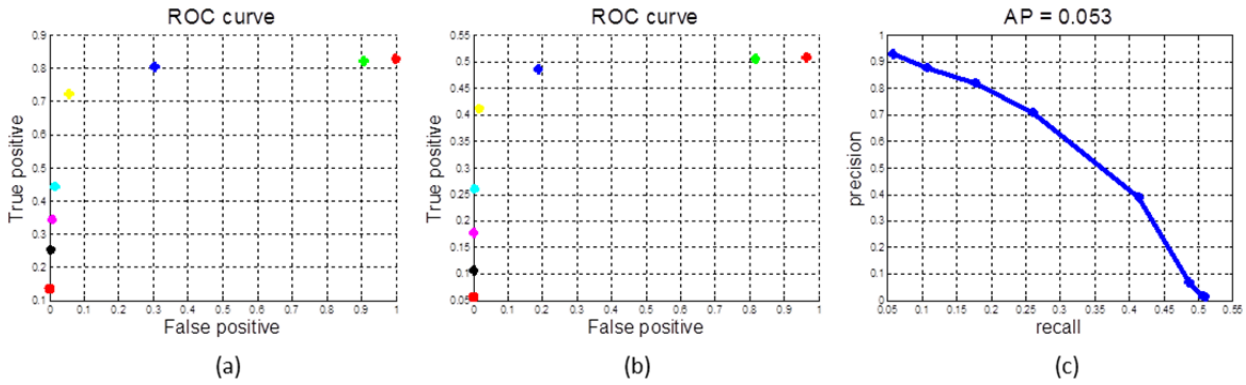


Figure 16. Results are shown for three approaches to car detection in MIT2317 photographs: ROC curve for the approach using the area (a), ROC curve using the number of detections (b), and precision-recall curve (c).

False-Alarm Filtering (heading level 1)

Geometric false-alarm filtering using 3D models requires both the position and pose of the camera taking the photograph and the geolocation of the candidate DPM detections of people and cars in order to place them in the prior 3D models. These two sets of information then allow us to perform line-of-sight calculations against buildings and checks of locations within known parking areas.

Mapping 2D DPM Detections into 3D World Space (heading level 2)

Photographs taken by conventional cameras represent 2D angle-angle projections of 3D world space into image planes. Once an image's camera position (georegistered) and pose are determined, the extrinsic and intrinsic parameters for its camera are known. Using a pinhole camera model (Figure 17), we can map a 3D point Q^{world} in world coordinates to a 2D point on the image plane, q, using the coordinate transformation

$$q = P_{3\times4} Q^{world},$$

where the projection matrix

$$P_{3\times4} \equiv K_{3\times3}[R|t]$$

has nine degrees of freedom. Upper-triangular $K_{3\times3}$ contains the camera's "intrinsic" parameters (one focal and two image plane center parameters); rotation $R_{3\times3}$ and translation $t_{3\times1}$ contain the camera's "extrinsic" parameters (three rotation and three translation parameters). This model ignores small distortions caused by the camera lens.

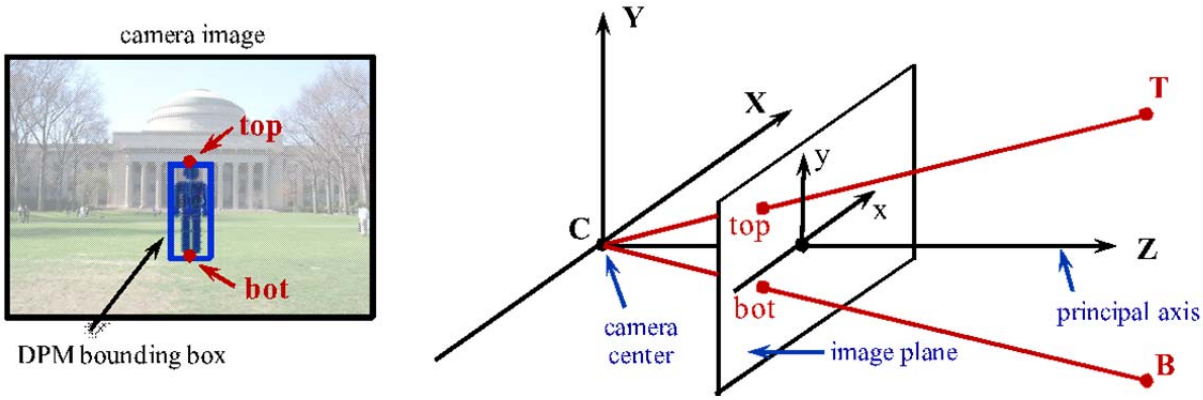


FIGURE 17. A pinhole camera model is used to map the top and bottom of a detected person in the 2D image plane into 3D space. Points top (*x, y*) and bot (*x, y*) are projected to T(X,Y,Z) and B(X,Y,Z), respectively.

The inverse mapping (projecting a 2D point on the image plane to a 3D point in world coordinates) requires an additional constraint. Each point in the image corresponds to a geometrical ray, and a 2D box in the image maps onto a 3D sub-frustum. In the absence of any length scales, distances to objects visible in an image, as well as their absolute sizes, are unknown. Indeed, Hollywood often takes advantage of this mathematical fact by filming miniatures in place of life-sized sets. But if the size of some object is a priori assumed, its range may simply be determined via backprojection, as shown in Figure 18. The average height for American adults is 1.7 meters. We therefore assumed that all standing people detected in our reconstructed MIT ground photos have this height. The top and bottom pixels of each 2D detection could then be projected into 3D space (local east, north, up [ENU] coordinates) by solving the simultaneous equations:

$$top = P_{3\times4}T^{world}$$

$$bot = P_{3\times4}B^{world}$$

$$T_U^{world} - B_U^{world} = h$$

$$T_N^{world} - B_N^{world} = 0$$

where h = 1.7 m. In the above equations, *top* and *bot* are the 2D pixel coordinates of the top and bottom of the detected person, respectively; $P_{3\times4}$ is the projection matrix from 3D world coordinates to 2D camera coordinates; $T^{world} = \begin{bmatrix} T_E^{world} & T_N^{world} & T_U^{world} & 1 \end{bmatrix}^{\mathrm{T}}$ and $B^{world} = \begin{bmatrix} B_E^{world} & B_N^{world} & B_U^{world} & 1 \end{bmatrix}^{\mathrm{T}}$ are the projected coordinates of the top and bottom of the detected person in 3D (ENU) space.

Note that, because of the constraints in the U and N dimensions, if the camera's *y*-axis is not parallel to the world *U*-axis, the resulting values of T_E^{world} and B_E^{world} will not be equal (Figure 19); however, the average of T_E^{world} and B_E^{world} can be taken to be the "center" of the person.

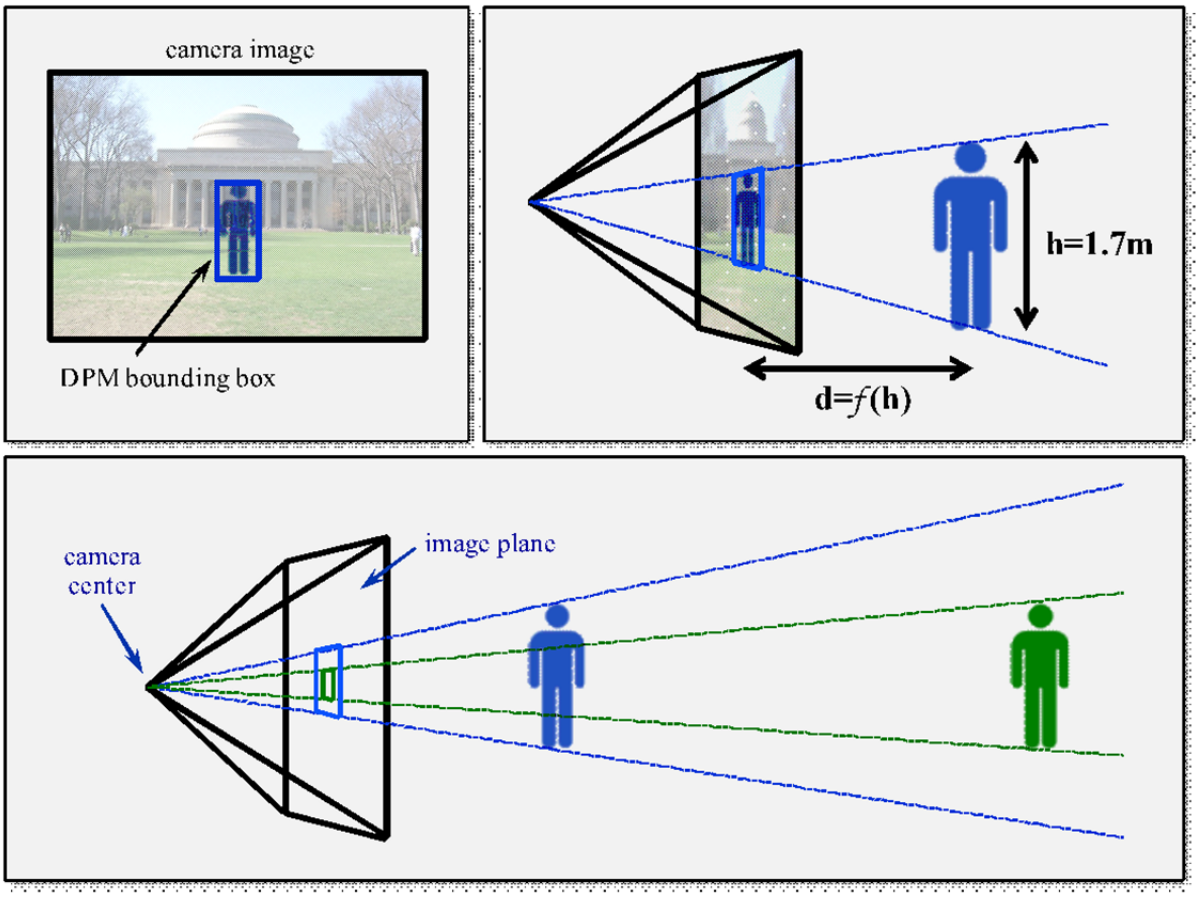


FIGURE 18. The range of an average-height person is determined via backprojection.

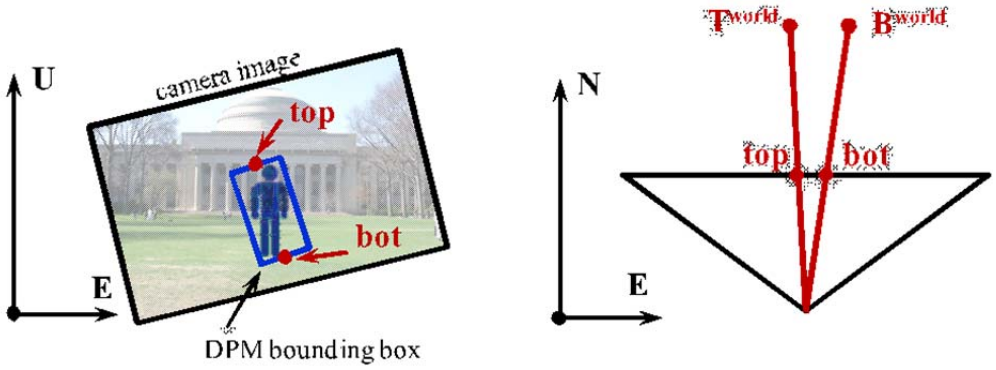


FIGURE 19. The figure illustrates the effect of the camera's *y*-axis not being parallel to the world *u*-axis.

Figure 20 illustrates an example of the geolocation technique as applied to an image from the MIT2317 dataset. The red box in the original image (Figure 20a) encloses one of several pedestrians walking outdoors nearby MIT East Campus. Figures 27b, 27c, and 27d illustrate the pedestrian's geolocation within the MIT map. While errors in image-plane placement of 2D bounding boxes, as well as 3D reconstruction, inevitably lead to uncertainties in human geolocation, it is reassuring to note that the pedestrian's world-*u* coordinate lies reasonably close to the ground plane.

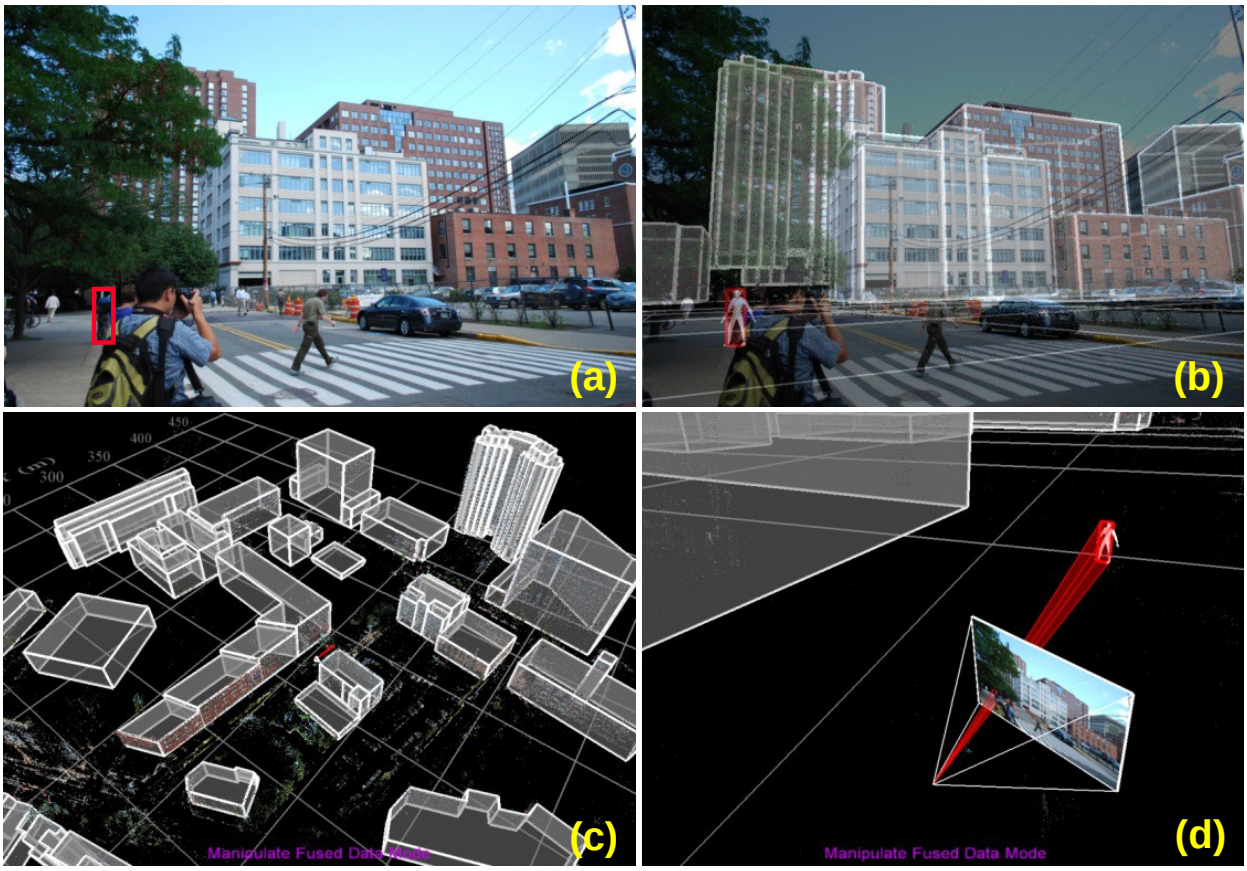


FIGURE 20. The images illustrate the process of person geolocation applied to a photograph from the MIT2317 dataset.

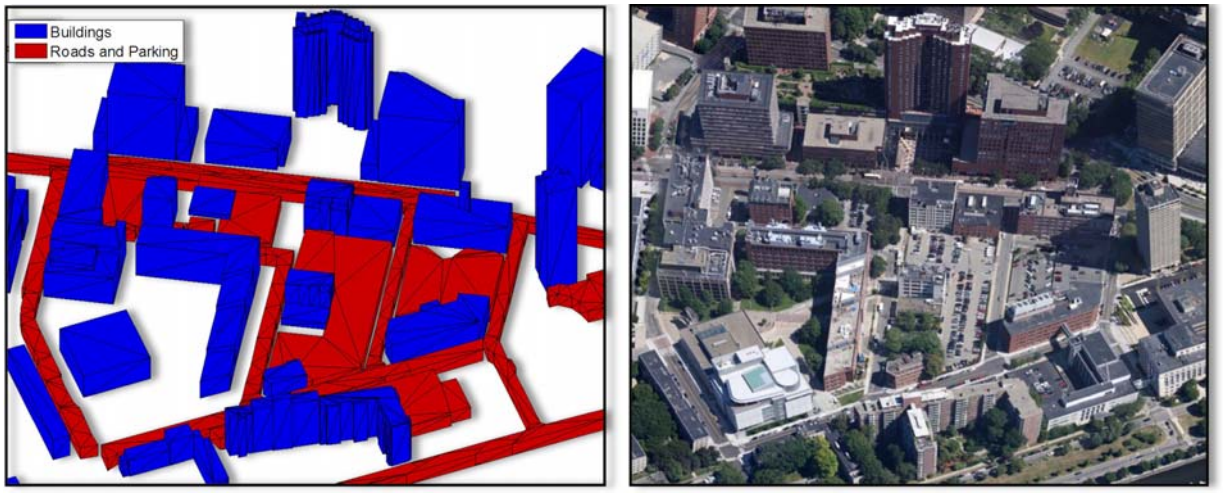
False-Alarm Filters (heading level 2)

Most genuine people observed in the MIT ground photographs backproject onto geopositions where their feet are close to solid ground. In contrast, false alarms from automatic human detection algorithms often yield 2D bounding boxes that map onto 3D geolocations floating in midair or submerged many meters underground. Alternatively, a backprojected human's location might pass through an opaque building wall. Geometry combined with common sense can consequently rule out such false alarms coming from automated object detectors. To decrease the number of false detections returned by the DPM detector, we built three false-alarm filters to leverage knowledge about the scene geometry from the previously built 3D world model:

- The HIGH filter rejects all detections that are projected into 3D space at a height greater than 5 meters above ground. As the MIT campus is approximately flat, the HIGH filter for our datasets

was a simple threshold on the projected person's elevation. However, this filter could also incorporate the world model by calculating the distance between the projected person's location and the surface below to take into consideration more complex terrain or to detect people standing on a building roof.

- The line-of-sight (LOS) filter is used to determine whether a detection is feasible given the building geometry in the scene. Once a candidate detection is projected into the 3D world, a ray is drawn from the camera location to the projected location. If the ray intersects a building face (or other solid obstacle such as the ground plane), the candidate detection is rejected, filtering out people that are seen "through" walls.
- The road and parking lot (ROAD) filter was used with the car-detection dataset only. Each candidate car detection is projected into the 3D world, and its location is checked against the model of road and parking lot locations (Figure 21). Detections not located on a road or parking lot are rejected as false positives.



(a) (b)

FIGURE 21. In (a), road and parking lots are highlighted in red, while building models are in blue. An aerial photograph of this same area is shown in (b).

Figures 22 and 23 show examples of the detection results after we applied the false-alarm filters. In Figure 22, of the two people in the scene, one is partially obscured and is not detected by DPM (boxed in blue). DPM returns about 20 detections (shown as green, yellow, and cyan boxes). Figure 23 shows where those detections are projected in the 3D world using a 1.7-meter height assumption. More than half of these detections were rejected by either the HIGH or LOS filters (colored in cyan); the remaining detections include one true positive (shown in green) and six false positives (yellow). By using the prior world model to filter out infeasible detections, the false-alarm rate was decreased by 68%.

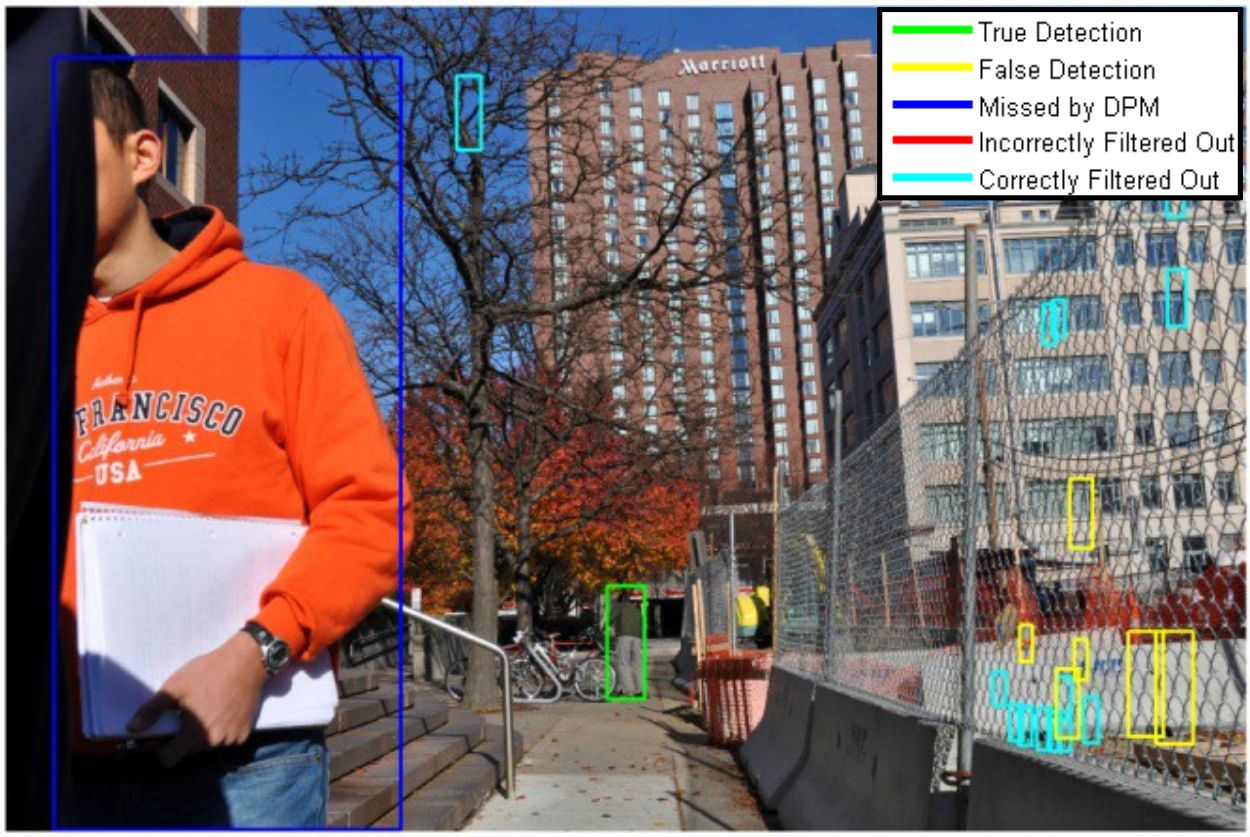


FIGURE 22. Detection results after applying the false-alarm filters are shown in the image space. About 20 detections were made. The one true detection is in the green box. The DPM algorithm missed the person on the left because he is partially obscured. The cyan boxes show the cases in which the false-alarm filters correctly eliminated those detections.

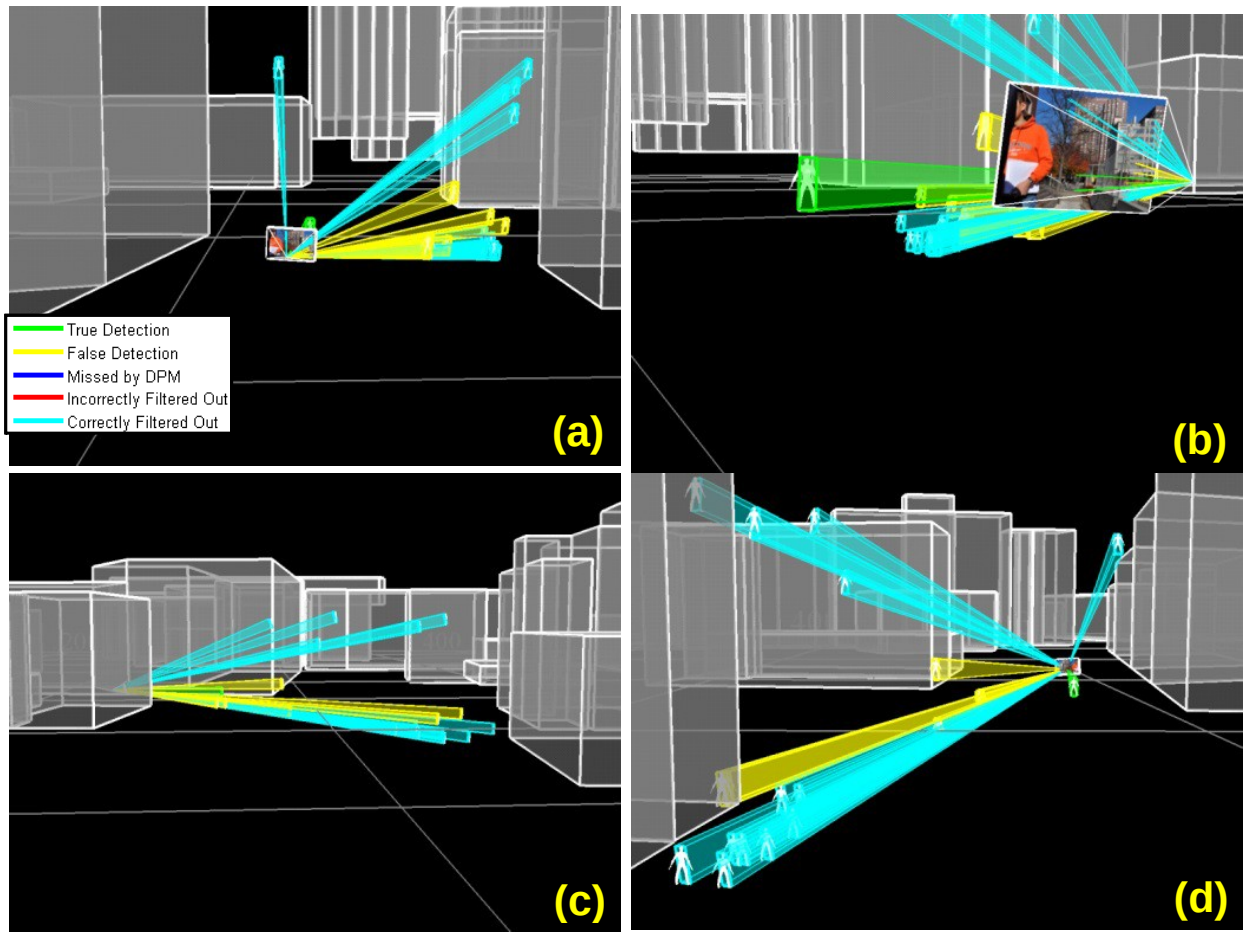


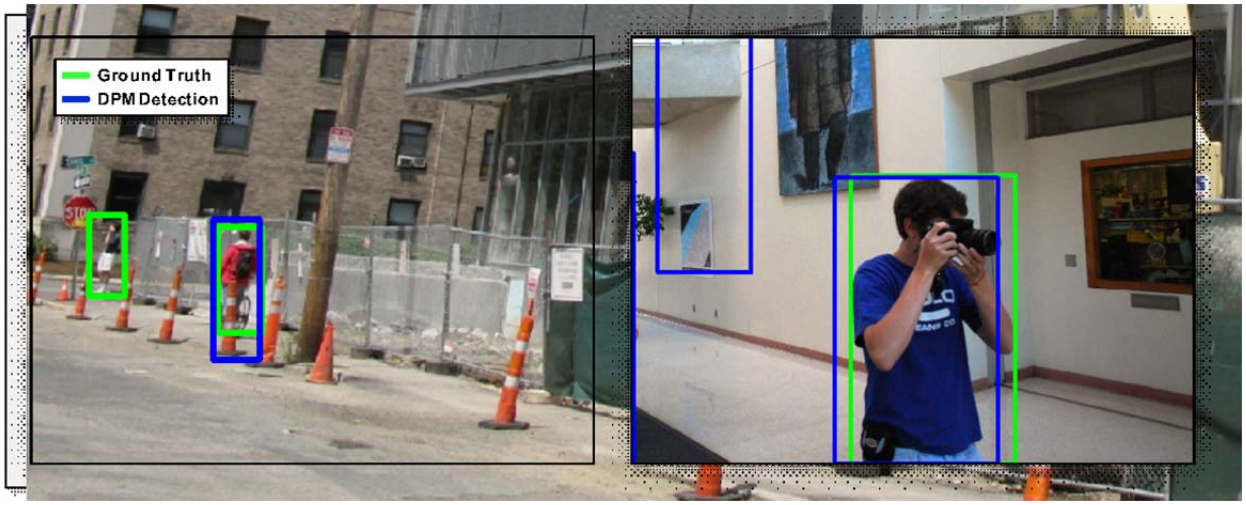
FIGURE 23. Figure 22's detection results after applying the false-alarm filters are shown in 3D space, with four different views depicted in (a)–(d).

Stochastic Extensions (heading level 2)

Our methods for projecting DPM detections into the 3D world and filtering out unlikely detections may produce errors (both false positives and false negatives) attributable to several factors:

- Inaccurate camera pose: A poor camera-pose estimate will cause detections to be projected to the wrong location. Any one of the false-alarm filters may then fail to reject a false positive, or worse, incorrectly reject a true detection. The LOS and HIGH filters are designed to be especially sensitive to accurate pitch-angle estimates because an error of even a few degrees in pitch may cause a true detection to be projected too high or below ground.
- Inaccurate world model: All three false-alarm filters depend on the world model being complete and accurate. If a parking lot is missing from the world model, for example, all cars detected in that parking lot will be rejected by the ROAD filter. Similarly, if a building is modeled in the wrong location or does not have the correct geometry, the LOS filter will erroneously accept or reject detections.
- Misaligned DPM detection box: Even when DPM successfully detects a person in the scene, the DPM bounding box may be slightly misaligned from the true bounding box (Figure 24a). A DPM

detection is considered a true positive if at least 50% of its bounding box area overlaps a ground-truth bounding box. However, even a slight misalignment (especially with respect to the person's height) may cause a significant shift in the projected location in the 3D world.



(a) (b)

FIGURE 24. In (a), a traffic cone is incorrectly detected as part of a person, causing the DPM bounding box to be taller than the ground truth. This detection is still considered a true positive, but the 3D projection may be significantly shifted. In (b), DPM makes a partial detection that qualifies as a true positive.

- Partial detections: DPM may detect partial people (for example, the torso only). Figure 24b shows an example of a true positive detection in which the bounding box stops at the person's knees. Since the bounding box does not surround the full height of the person, the 1.7-meter height assumption will cause the detection to be projected farther away, making it more likely to be projected "through" a wall or the ground plane.
- Invalid height assumption: Even in the absence of any sources of error, 2D detections are projected into the 3D world under the assumption that people are 1.7 meters tall. This supposition is obviously not true for all people and may cause even perfect DPM detections to be projected to the wrong location.

In an effort to reduce the effects of some of the above problems, a stochastic false-alarm filter was developed. Rather than projecting detections deterministically, the 3D projection is expressed as a nonlinear function *g*:

$$[x_{E,det}, x_{N,det}, x_{U,det}] = g(top, bot, h, x_{E,cam}, x_{N,cam}, x_{U,cam}, \theta_{cam}, \varphi_{cam}, \psi_{U,cam}, f, c_1, c_2),$$

where *top* and *bot* are the pixel values of the top and bottom of the DPM detection box; h is the person's height; $x_{E,cam}, x_{N,cam}, x_{U,cam}, \theta_{cam}, \varphi_{cam}, \psi_{U,cam}$ are the camera pose values corresponding to 6–degree of freedom position and rotation; and f, c_1, c_2 are the camera's intrinsic parameters. These 12 variables

can be sampled from a prior distribution that takes into consideration uncertainties in factors such as camera pose and person height. The resulting calculations of $x_{E,det}, x_{N,det}, x_{U,det}$ are Monte Carlo samples of the person's projected location distribution in the 3D world. If, for a given DPM detection, the number of samples that pass all false-alarm filters exceeds some predefined threshold, the detection is accepted. By setting a low threshold, true detections are less likely to be incorrectly filtered out because of small pose errors or an incorrect height assumption, but the false-alarm rate will also be higher. This stochastic method will not correct for gross errors in the world model (for example, a new building is not present), but it is more robust to small projection inaccuracies.

Figure 25 illustrates how this sample-based approach can be used to relax the height assumption for detected people. By sampling *h* from a truncated normal distribution, the DPM detection is projected out to multiple distances from the camera. Even if some of the sampled projections are rejected by a filter (in this case, the LOS filter would reject samples projected through a building wall), the detection may still be accepted if some of the samples pass.

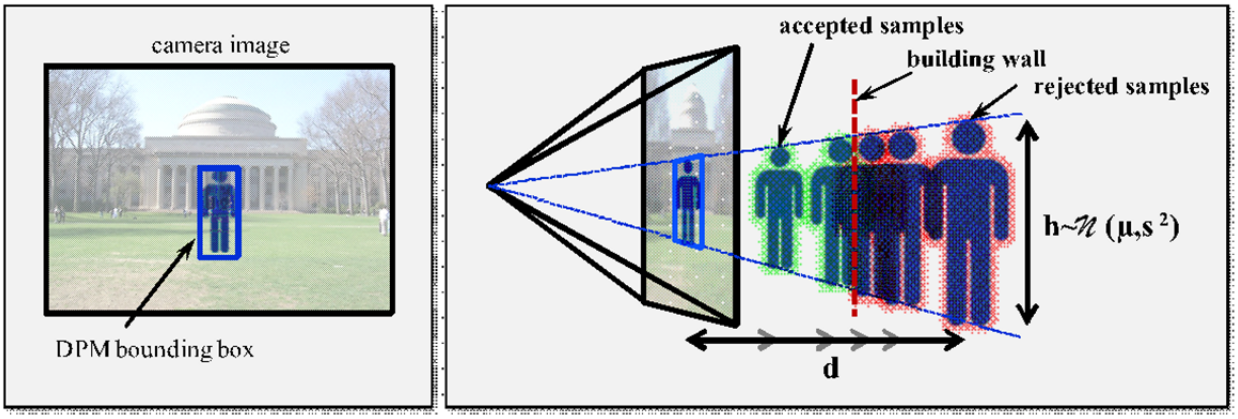


FIGURE 25. A sample-based approach allows for the relaxation of the height assumption.

People Detection Analysis and Results (heading level 2)

Figures 26 and 27 show ROC and precision-recall curves for people detection using the LOS and HIGH filters. For all cases presented here, the HIGH filter successfully filters out false positives without sacrificing the performance of true-positive detection. The LOS filter, which rejects detections that are projected through solid surfaces such as buildings or the ground plane, appears more sensitive to model parameters such as the assumed human height. Note that in the ROC curves, the horizontal axis is the average number of false alarms per image, which is a false-alarm rate rather than a probability of false alarm.

Figure 26 compares the baseline DPM results with those obtained from applying the LOS filter, HIGH filter, and both LOS and HIGH filters, using a deterministic 1.7-meter height estimate for projecting detections into 3D space [19]. For lower DPM confidence thresholds ($t_{DPM} \leq 0$), the LOS filter decreases the number of false alarms but also rejects some true positives. The HIGH filter further rejects false alarms without rejecting any true positives. Figure 26(b) shows a zoomed-in view of the ROC curve (representing the operating point near the "knee" of the curve). Here, the application of the LOS and

HIGH filters reduces the false alarms by 35% and the true positives by just under 1%. Because true detections are being wrongly rejected, we conjecture that the degraded performance is due to one or more of the factors described in the Stochastic Extensions section.

One factor that could contribute to the rejection of true positives is that many ground-truthed people in the MIT2317 image set are not completely visible from head to toe. When DPM detects these "partial people" (as it is designed to do), the 1.7-meter height assumption used to project the DPM bounding boxes into the 3D world is inaccurate. A bounding box around a partial person (for example, just a torso) is too short and will cause the detection either to be projected farther away from the camera or to be projected through a building wall or high above the ground plane.

To increase the robustness of the filters, we implemented a stochastic projection model as described in the Stochastic Extensions section. For each candidate detection, 20 height values were randomly sampled from a truncated normal distribution of American adult heights $\sim N(1.7,\ 0.16^2)$ [19], projected into 3D space, and either accepted or rejected by the LOS and HIGH filters. The resulting sample set is a discrete approximation of an "acceptance distribution," p_a, defining the probability that the detection is feasible. By setting a threshold on p_a, we can control the degree to which "corner-case" detections (e.g., a person standing very near a wall) are accepted or rejected.

Figure 27 shows results for one run of the stochastic algorithm. Candidate detections are accepted if any sample passes the filter, approximating a 5% threshold on p_a. This strategy is likely to retain detections (both true detections and false alarms). For this set of results at $t_{DPM} = -0.5$, applying the combination of LOS and HIGH filters decreases the false-alarm rate by approximately 29% over the baseline DPM detections without throwing out any true positives.

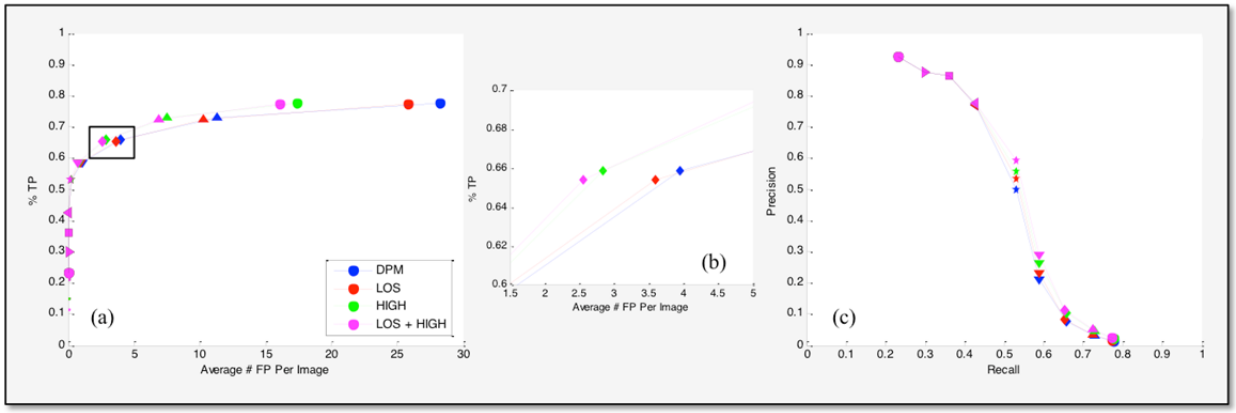


FIGURE 26. ROC (a) and precision-recall (c) curves are shown for human detection using a deterministic 1.7-meter height estimate. A zoomed-in ROC curve is shown in (b).

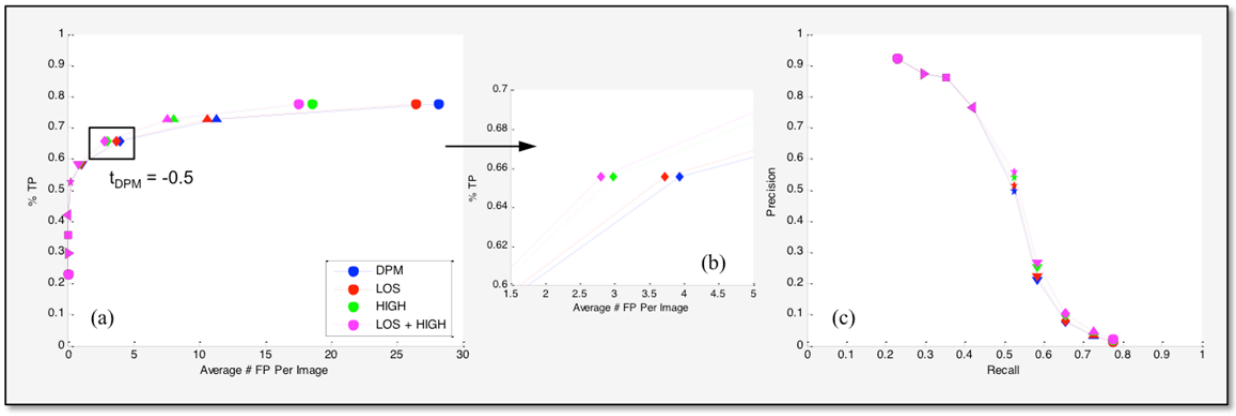


FIGURE 27. ROC (a) and precision-recall (c) curves are shown for human detection with candidate detections accepted if any samples pass the filter. A zoomed-in ROC curve is shown in (b).

Car Detections Analysis and Results (heading level 2)

Figures 28 and 29 show results for detections of cars using the LOS and HIGH filters, plus the additional ROAD filter, which only accepts cars whose projected locations are along a modeled road or in a parking lot. Unlike the DPM human detector, the car detector is only trained on full cars; therefore, for the results presented here, only fully visible cars were included in the ground-truth set.

Figure 28 illustrates the performance of the three false-alarm filters for the deterministic case (all cars are projected with an assumed height of 1.5 meter). At $t_{DPM} = -0.75$ (the knee on the ROC curve), the LOS filter rejects about 1% of true positives detected by DPM, and the ROAD filter rejects almost 9%. One could argue that the most likely cause of the rejection of true positives is an incorrect height assumption. In fact, a compact car or convertible may be closer to 1.2 meters tall, whereas a minivan or sport utility vehicle (SUV) may be around 1.8 meters tall. Next, a height distribution of $\sim N(1.5,\ 0.3^2)$ was used to project the car detections into the 3D world. Using a low threshold ($p_a \geq 5\%$), the stochastic LOS filter no longer rejects any true positives (results shown in Figure 29). At $t_{DPM} = -0.75$, all three filters combined reject only about 1% of true positives while decreasing the false positive rate 15% over DPM. At even lower DPM confidence thresholds, the performance increase is even more drastic: at $t_{DPM} = -1.25$, false detections are decreased by nearly 70% and true positives are rejected only 1% by applying all three false-alarm filters.

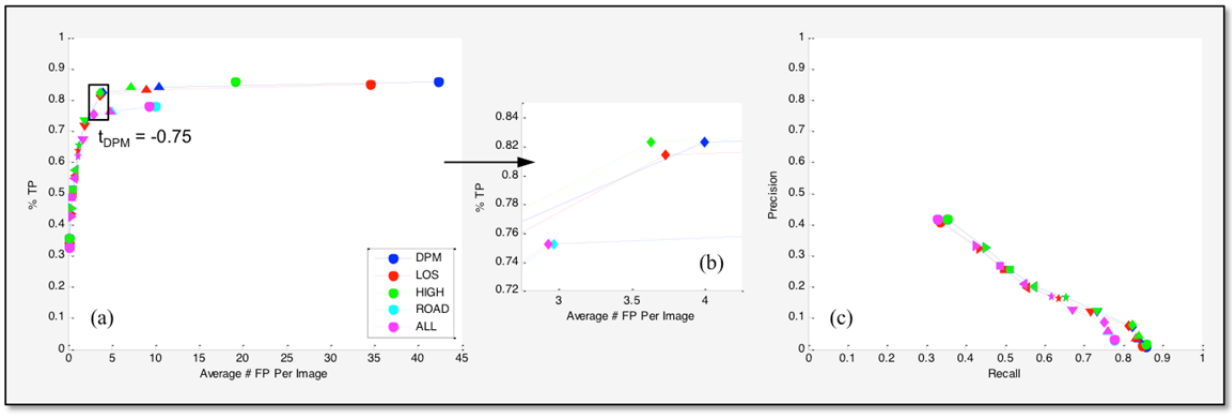


FIGURE 28. ROC (a) and precision-recall (c) curves are shown for car detection using a deterministic 1.5-meter height estimate. A zoomed-in ROC curve is shown in (b).

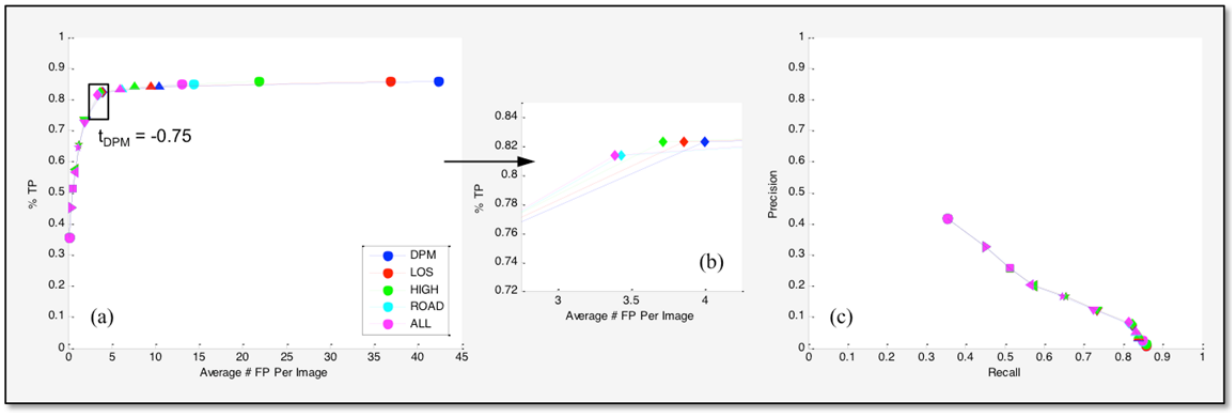


FIGURE 29. ROC (a) and precision-recall (c) curves are shown for car detection using a stochastic model with 5% acceptance rate. A zoomed-in ROC curve is shown in (b).

Conclusions and Future Directions (heading level 1)

The use of a prior 3D world model for geometric false-alarm filtering can eliminate ~30% to 70% of false alarms produced by the DPM object-detection algorithm, with minimal to no loss of true detections. This approach is especially useful for detection problems in which a high cost is associated with missing a true detection and resources must be expended to deal with the associated high level of false alarms. For mobile robot applications in a netcentric environment, 3D world models built and maintained by offboard sensor systems could improve local object-detection performance.

Many other potential improvements to local detection algorithms could leverage information stored in a 3D world model. Depending on the type of object to be detected, additional semantic models could be defined, analogous to the roads and parking lots for detecting cars. For example, detecting and reading signs could be improved by a model that designates the likely places to find signs, such as the edges of roads and on the sides of buildings.

Adding information on surface reflectances to the 3D model surfaces, such as color and texture, could simplify the the segmentation of images for change detection. This revised 3D model could be used to

predict the view a local camera should see based on the latest 3D model and could help identify the things in the world that have not changed, allowing additional processing to focus on potential changes and moving objects.

The semi-manual process for building and maintaining the 3D models of buildings, roads, and parking lots would need to be automated for an operational system. The authors envision an automated system in which newly collected sensor data from airborne, spaceborne, and ground sensors (including those on autonomous vehicles) would be compared to the latest 3D world model, and changes would then made to the model as appropriate. Identifying the data that contain new information and flagging them for additional analysis could help solve today's data-glut problem in which most sensor data that are collected are not processed or analyzed. Finally, netcentric technologies for federating multiple 3D models storing different aspects of the world (e.g., semantic, reflectance) would need to be developed so that a given type of detection algorithm can efficiently access only the model data that it needs.

REFERENCES

1. P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, 2010, pp. 1627–1645.
2. P.F. Felzenszwalb, R.B. Girshick, and D. McAllester, Discriminatively Trained Deformable Part Models, Release 4, available online at http://cs.brown.edu/~pff/latent-release4/.
3. D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, 2004, pp. 91–110.
4. N. Snavely, S.M. Seitz, and R. Szeliski, "Photo Tourism: Exploring Photo Collections in 3D," *Proceedings of the ACM (Association for Computing Machinery) SIGGRAPH (Special Interest Group on Graphics and Interactive Techniques) Conference*, 2006, pp. 835–846.
5. Bundler: Structure from Motion (SfM) for Unordered Image Collections, available online at http://phototour.cs.washington.edu/bundler/.
6. C. Wu, "VisualSFM: A Visual Structure from Motion System," available online at http://www.cs.washington.edu/homes/ccwu/Visual-SFM/.
7. Z. Sun, N. Bliss, and K. Ni, "A 3D Feature Model for Image Matching," *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 2194–2197.
8. K. Ni, Z. Sun, and N. Bliss, "3D Image Geo-registration Using Vision-Based Modeling," *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 1573–1576.
9. H. Viggh and K. Ni, **"**SIFT-Based Localization Using a Prior World Model for Robot Navigation in Urban Environments," *Proceedings of the 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing*, available at WorldComp online at http://worldcomp-proceedings.com/proc/p2012/IPC3329.pdf.
10. The PASCAL Visual Object Classes Homepage, http://pascallin.ecs.soton.ac.uk/challenges/VOC/.

11. P. Cho and N. Snavely, "3D Exploitation of 2D Ground-Level and Aerial Imagery," *Proceedings of the 2011 IEEE Applied Imagery Pattern Recognition Workshop*, 2011, pp. 1–8.
12. A.Y. Ng, M.I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems*, vol. 14, 2001, pp. 849–856.
13. S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.
14. K. Grauman and T. Darrell, "The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features," *Proceedings of the 10th IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1458–1465.
15. Spatial Pyramid code available at S. Lazebnik website, Department of Computer Science, University of Illinois at Urbana-Champaign, www.cs.illinois.edu/homes/slazebni/.
16. Massachusetts Office of Geographic Information (MassGIS), http://www.mass.gov/anf/research-and-tech/it-serv-and-support/application-serv/office-of-geographic-information-massgis/.
17. MATLAB® CENTRAL, File Exchange for Alpha Shapes by J. Lundgren, available online at http://www.mathworks.com/matlabcentral/fileexchange/28851-alpha-shapes.
18. M. Everingham and J. Winn, The PASCAL Visual Object Classes Challenge 2010 (VOC2010) *Development Kit, available at http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/#devkit*.
19. M.A. McDowell, C.D. Fryar, C.L. Ogden, and K.M. Flegal, "Anthropometric Reference Data for Children and Adults: United States, 2003–2006," National Health Statistics Reports, Number 10, October 22, 2008.