The Semantic Paintbrush: Interactive 3D Mapping and Recognition in Large Outdoor Spaces

Ondrej Miksik[1†⋆] Vibhav Vineet[2◇⋆] Morten Lidegaard[1†] Ram Prasaath[◇] Matthias Nießner[2]
Stuart Golodetz[1†] Stephen L. Hicks[1‡] Patrick Pérez[3] Shahram Izadi[4] Philip H. S. Torr[1†]
[1]University of Oxford [2]Stanford University [3]Technicolor R&I [4]Microsoft Research

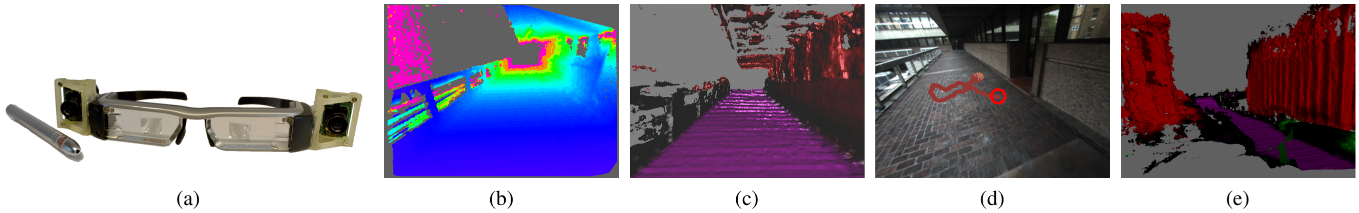

Figure 1: (a) Our system comprises of an off-the-shelf pair of optical see-through glasses, with additional stereo RGB-Infrared cameras, and an additional handheld infrared/visible light laser. (b) The passive stereo cameras are used for extended range and outdoor depth estimation. (c) The user can see these reconstructions immediately using the heads-up display, and can use a laser pointer to draw onto the 3D world to semantically segment objects (once segmented these labels will propagate to new parts of the scene). (d) The laser pointer can also be triangulated precisely in the stereo infrared images allowing for interactive 'cleaning up' of the model during capture. (e) Final output, the semantic map of the scene.

ABSTRACT

We present an augmented reality system for large scale 3D reconstruction and recognition in outdoor scenes. Unlike existing prior work, which tries to reconstruct scenes using *active* depth cameras, we use a purely passive stereo setup, allowing for outdoor use and extended sensing range. Our system not only produces a map of the 3D environment in real-time, it also allows the user to draw (or 'paint') with a laser pointer directly onto the reconstruction to *segment* the model into objects. Given these examples our system then learns to segment other parts of the 3D map during online acquisition. Unlike typical object recognition systems, ours therefore very much places the user 'in the loop' to segment particular objects of interest, rather than learning from predefined databases. The laser pointer additionally helps to 'clean up' the stereo reconstruction and final 3D map, *interactively*. Using our system, within minutes, a user can capture a full 3D map, segment it into objects of interest, and refine parts of the model during capture. We provide full technical details of our system to aid replication, as well as quantitative evaluation of system components. We demonstrate the possibility of using our system for helping the visually impaired navigate through spaces. Beyond this use, our system can be used for playing large-scale augmented reality games, shared online to augment streetview data, and used for more detailed car and person navigation.

Author Keywords

3D reconstruction, laser pointer interaction, semantic segmentation, stereo, augmented reality, visually impaired

INTRODUCTION

Maps help us to navigate and discover the world. In recent times, companies such as Google and Microsoft have applied reconstruction techniques to aerial and/or street-level imagery to produce virtual 3D maps on a *global scale*. These digital 3D maps form the basis of many of the navigation systems we use in our cars and mobile devices today.

Whilst much progress has been made in 3D mapping, particularly with the advent of real-time depth cameras, most of these virtual maps are still at a *geometric* level, representing the 3D structure of the scene, as opposed to understanding or *recognizing* the higher level objects or scene structure. Furthermore, these maps are captured ahead of time, and often at a coarse level, instead of being captured *live* and reflecting the detailed nature of the scene. Finally, these maps are often general-purpose and not *personalized* to specific objects or areas of interest for specific users.

In this paper, we present a new mapping system that is capable of creating large-scale semantic maps of outdoor scenes *interactively*. The word 'interactive' is of particular importance, as this not only implies live capture of the map, but also a system that keeps the user 'in the loop' to guide the mapping towards objects and elements of the map that are of particular interest.

More specifically, we present a novel augmented reality (AR) system for large scale 3D reconstruction and recognition in outdoor scenes. Unlike prior work, which tries to reconstruct scenes using *active* depth cameras, we use a purely passive stereo setup, allowing for outdoor use and extended range sensing. This allows us to reconstruct large and/or distant structures, such as building facades, roads and cars.

Our system not only produces a map of the 3D environment in real-time, it also allows the user to draw (or 'paint') with a laser pointer directly onto the reconstruction. The user simply points at an object with the laser pointer, performs a brush-like stroke, and then issues a voice command to interactively segment and label the 3D scene into different object classes. Unlike typical object recognition systems, which work in a 'closed-world' scenario, with a fixed, pre-trained set of object

[⋆] O. Miksik and V. Vineet assert joint first authorship.
[†] Department of Engineering Science
[‡] Nuffield Department of Clinical Neurosciences
[◇] Work was done at the University of Oxford.

ondra.miksik@gmail.com, vibhav.vineet@gmail.com | http://www.miksik.co.uk

CHI '15, Apr 18 – 23 2015, Seoul, Korea
Copyright 2015 ACM 978-1-4503-3145-6/15/04$15.00.
http://dx.doi.org/10.1145/2702123.2702222

classifiers, our system is fully interactive, allowing the user to add new classes on the fly, and even correct object labels.

The laser pointer is additionally triangulated by the stereo camera rig during capture, which provides a strong 3D prior to help *interactively* 'clean up' the stereo reconstruction and final 3D map. Stereo algorithms typically break in textureless regions, causing major errors. Here, these errors can be quickly and interactively cleaned up by the user, in an online manner. To our knowledge, this is the first such system that allows the user to see the results of object and stereo estimation in real time and interactively correct them.

With our system, within minutes, a user can capture a full 3D map, segment it into objects of interest and refine parts of the model during capture, all by simply exploring the space and moving a handheld laser pointer device, metaphorically 'painting' or 'brushing' onto the world. We provide full technical details of our system to aid replication, as well as quantitative evaluation of system components.

We are particularly interested in application scenarios that can exploit these large-scale semantic 3D maps. We demonstrate the possibility of using our system for helping the visually impaired navigate through spaces. Here, the semantic segmentation allows us to highlight objects of interest using the AR glasses. The metric and precise reconstruction can be used for navigation, and the laser pointer can be used to pinpoint objects within proximity. Beyond this use, these semantic maps can used for playing large-scale AR games, shared online to augment streetview data, and used for more detailed car and person navigation.

Our contributions can therefore be summarized as follows:

- A novel augmented reality hardware system comprising of transparent LED glasses, attached RGB-Infrared stereo cameras, and a one-button laser pointer.
- A large-scale dense mapping system that can operate in outdoor scenes, in real-time. This provides the ability to reconstruct objects at greater distances and in direct sunlight, beyond the capabilities of active depth cameras such as the Kinect.
- Extensions to KinectFusion [22] to support visual odometry for pose estimation alongside stereo data input.
- The ability for users to semantically segment captured 3D maps into object regions using a simple laser pointer and 'brushing' metaphor.
- A machine learning pipeline for learning from these object examples to automatically segment the captured 3D models, in real-time, at scale, and with noisier data than previous systems *e.g.* [32].
- Integration of accurate yet sparse measurements from a laser pointer to interactively improve the quality of the stereo depth estimation and reconstruction.
- A first prototype of our semantic mapping system for the visually impaired.

RELATED WORK

In the past years, there have been rapid developments in algorithms and systems for indoor and outdoor mapping, at varying scales. Offline structure from motion (SfM) and multi-view stereo (MVS) techniques work directly on photos taken of the same scene from different viewpoints (potentially from online repositories and heterogeneous sets of cameras). These systems typically utilize computationally expensive feature matching and bundle adjustment techniques and require minutes, hours or even days to create 3D models. The output can be sparse [31] or dense point clouds [1], or even detailed and connected surface models [9].

Algorithms based on Simultaneous Localization and Mapping (SLAM) [18, 4] instead perform real-time mapping using a single monocular cameras. Early on, they represented the world by a small number of reconstructed 3D points. With the advent of *dense* real-time methods such as DTAM [23], they have moved to reconstructing detailed surfaces. However, they can still only reconstruct very small environments.

Recently LSD-SLAM [5] demonstrated large-scale semi-dense point cloud reconstruction using only a monocular mobile phone camera. The method is based on a variant of semi-dense whole image alignment for camera tracking [6]. Other notable systems focusing on city-scale reconstructions use passive cameras (*e.g.* Taneja *et al.* [30]). Chen *et al.* [2] localizes landmarks at city-scale on mobile devices. Geiger *et al.* [10] use stereo camera input to build a dense 3D reconstruction of scene in real-time.

Another approach is to replace the challenges of depth estimation using passive cameras with the use of active depth sensors, such as structured light or time-of-flight systems. The ability to compute (noisy) real-time depth maps cheaply has led to a resurgence of dense 3D reconstruction algorithms, demonstrating real-time performance [22] even at large scales [25], [34]. However, these systems rely on active sensors, which limits their use outdoors (*i.e.* in direct sunlight or at extended sensing ranges). The ability of such systems to reconstruct objects such as buildings at long-range is thus limited.

Whilst these systems have demonstrated impressive 3D mapping results, they stop purely at geometry reconstruction. Instead another important area is the recognition of scene objects. A great deal of work has focused on developing efficient and accurate algorithms predicting object labels at the pixel level. Examples include the models of Ladicky *et al.* [19] or Munoz *et al.* [21]. Recently many others have focused on labeling *voxels* or other 3D representations. Some of them focus on indoor scenes [33], and others on outdoor scenes [29], [36]. Some other recent works have also tried to jointly optimize for both the tasks of reconstruction and recognition, and so incorporate the synergy effects between these two high level vision tasks [12].

However, these methods are far from real-time and require offline processing. A more recent trend has explored real-time or online object recognition directly during 3D mapping [28], in particular to help compress 3D models further and aid in relocalization. SemanticPaint [32] takes this concept further by allowing users to label the scene during capture by *touching* surfaces and providing an online learning framework to infer class labels for unseen parts of the world. We build on that framework in this paper, but change the algorithm to handle large scale outdoor mapping using far nosier stereo data. Furthermore, we fundamentally change the input modality. Using a laser pointer allows for the correction of failures in the estimated stereo depth, which are endemic when moving to outdoor scenes. This ability to create semantic segmentations of the map and correct both the geometry and labels is a critical part of our system.

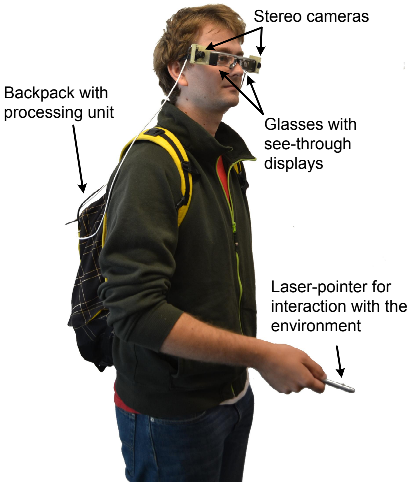
Stereo cameras
Backpack with processing unit
Glasses with see-through displays
Laser-pointer for interaction with the environment

Figure 2: The main hardware components of our system.

Laser pointers have been used extensively for HCI scenarios, especially for interacting with large displays in intuitive ways [35, 26, 17]. Our approach uses the laser pointer as a means for interactively segmenting the scene into objects of interest. However, as a by-product the laser pointer can be precisely triangulated through the stereo pair. This allows us to also refine the 3D model, based on accurate 3D measurements taken from the laser pointer. This is in the spirit of systems such as [11] and other scanning laser depth sensors. However, our method puts the user 'in the loop' allowing the laser to refine parts that the user cares about or observes as noisy. In contrast to [24], our method allows to label any unknown (indoor/outdoor) environment into objects and semantic parts.

We demonstrate our real-time large scale-semantic mapping system in the context of helping visually impaired users to navigate through outdoor spaces. Here, laser pointing devices have a long history as digital aids for the partially sighted (see [16] for a review). We however take further inspiration from a relatively new trend of helping the visually impaired through augmented vision [14, 8]. The basic principle is based on capturing images using a regular camera or depth sensor, and enhancing features of the image such as edges [8] or objects of interest [14]. These enhanced images are then displayed to the user on head-mounted AR-glasses, hence stimulating the residual vision of the user.

SYSTEM OVERVIEW

In this section, we describe our system from a hardware, user interaction, and software perspective.

Hardware

The hardware for our system is shown in Fig. 2 and Fig. 3. It is composed of optical see-through AR glasses (EPSON MOVERIO BT-200) with a resolution of 960×540 and field of view of approx. 23° corresponding to a 40" virtual screen at 2.5 metres. Attached to these glasses are a pair of Omnivision RGB-Infrared (RGB-I) cameras (OV4682 RGB IR) with a resolution of 2688x1520 pixels. These cameras are capable of imaging both visible and infrared (IR) spectra. The cameras are set apart with a baseline of 22cm, calibrated, and using a stereo algorithm (described in the next section), natural features in the RGB image of the left camera are *matched* with those in the right, to estimate the disparity of the scene. This allows a *dense* depth map to be computed per frame. Additionally, we use a standard red laser operating at visible and IR spectra between 680-730nm, with output far less than 5mW making its usage eye safe. This laser emits both red light for the user to see, but also IR light which can be sensed by the IR sensitive pixels of the stereo cameras. This laser point can be triangulated and used to help localize the pointer with respect to the 3D reconstruction.

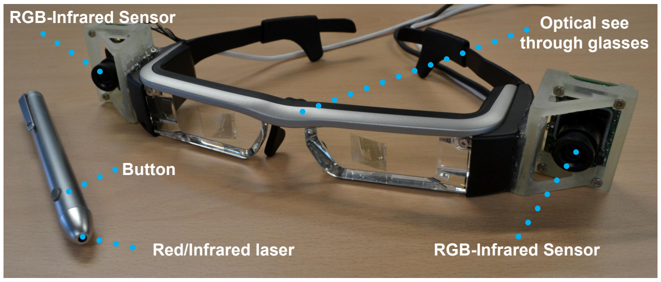

Figure 3: The main hardware components of our augmented reality glasses. See text for details.

User Interface and Interaction

As the user wears the AR glasses, they are provided with immediate feedback, as the reconstruction is captured live. The reconstruction is based on a scalable variant of the KinectFusion system [25]. Our system not only produces a map of the 3D environment in real-time, it also allows the user to draw (or 'paint') with a laser pointer directly onto the reconstruction.

In Fig. 9 and the accompanying supplementary video, we show how the laser pointer is used for interaction. In a basic scenario, a user wears the AR glasses and carries the laser pointer and backpack with processing unit (see Fig. 2). The user simply points at an object with the laser, performs a stroke, and then issues a voice command to interactively segment and label the 3D scene into different object classes. Unlike typical object recognition systems, our system therefore very much places the user 'in the loop' to segment particular objects of interest, rather than learning from predefined databases. The laser pointer is additionally triangulated by the stereo camera rig during capture, which provides a strong 3D prior to help 'clean up' the stereo reconstruction and final 3D map, *interactively*. Stereo algorithms typically break in textureless regions, causing major errors and outliers. Here, these errors can be quickly and interactively cleaned up by the user, in an online manner. The immediate feedback is visualized on the AR glasses. Our system also supports multi-user interactions, this scenario is thoroughly discussed in the Applications section.

SOFTWARE PIPELINE

The entire pipeline consists of several steps (*cf.* Fig. 4). First, we capture a pair of frames from the RGB-I cameras, which we separate into a pair of color and IR images. Next, we estimate the depth and camera pose from the color images, and detect and track the laser dot in the IR images. Then, the system fuses the input data to the current 3D model and performs 3D inference to propagate the semantic information and improves the reconstruction by high quality depth from triangulated laser dots. The user interacts with the system at all stages, by moving with the wearable AR device and by laser pointer to label the objects and improve the depth. The output of the system is continuously visualized with the optical see-through glasses.

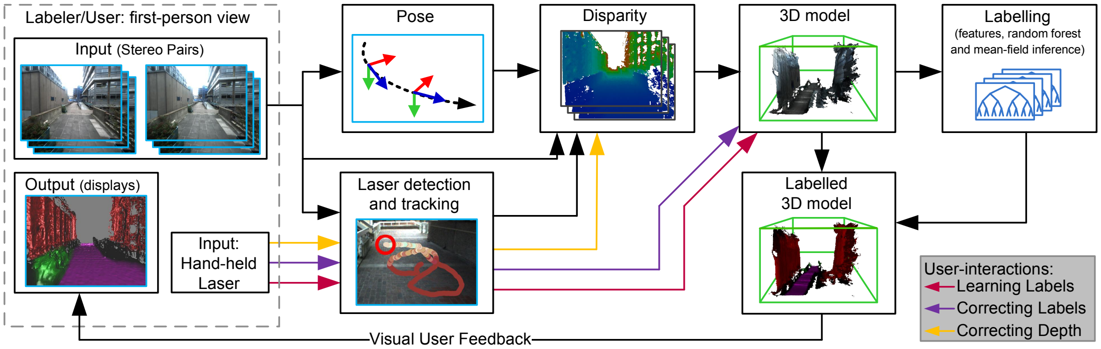

Figure 4: Overview of the system.

We first describe the more standard or known components of our system to aid replication, before moving onto the more novel and interactive aspects.

Surface Reconstruction The depth data generated using the stereo pairs are generally noisy. In order to generate high-quality surfaces, we follow the scalable hash-based fusion approach of Nießner *et al.* [25]. The key property of this approach is that it is able to generate high-quality surfaces of large-scale indoor scenes by fusing noisy depth data measured over time. However, there are two main drawbacks of this system: 1) the system is fully dependent on Kinect data, hence it fails to work in an outdoor environment and 2) the method depends on the ICP approach for camera pose estimation. Their pose estimation system generally fails with noisy depth from stereo. In our work, the first key contribution is to adapt this scalable hashing to work with outdoor scenes given stereo pairs and also to solve the issues associated with camera tracking.

Given the noisy depth data generated using stereo image pairs and pose estimate obtained by the visual odometry (described later), we incrementally fuse them into a single 3D volume using a truncated signed distance function (TSDF) [3] within a volumetric data structure. Each voxel is represented by a color, the TSDF encoding the distance of the voxel to the closest surface interface, and a weight that measures the confidence of that voxel belonging to the surface.

Camera calibration

First an offline calibration process is performed on the two RGB-I cameras. This comprises of: 1) *intrinsic calibration* to compute the geometric parameters of each IR camera lens (focal length, principal point, radial and tangential distortion); 2) *stereo calibration* to compute the geometric relationship between the two cameras, expressed as a rotation matrix and translation vector; 3) *stereo rectification* to correct the camera image planes to ensure they are scanline-aligned to simplify disparity computation. For more details please see [13].

Depth Estimation

Given a stereo image pair, we compute the depth as $z_i = bf/d_i$ where z_i is the depth for a corresponding disparity value d_i of i^{th} pixel, b is the stereo camera baseline and f is the camera focal length. For disparity estimation, we use an approach of Geiger *et al.* [10] which forms a triangulation on a set of support points that can be robustly matched. This reduces the matching ambiguities and allows efficient exploitation of the disparity search space without any global optimization. Hence, the method can be easily parallelized.

Visual Odometry

For camera pose estimation, we use the FOVIS feature-based visual odometry method [15]. First, we preprocess an input pair of images by spatially smoothing them with a Gaussian filter and building a three-level image pyramid, in which each pyramid level corresponds to one octave in scale space. Then, we extract a set of sparse local features representing corner-like structures. For this, we use a FAST detector [27] with an adaptively-chosen threshold to detect a sufficient number of features. The feature extraction step is "biased" by bucketing to ensure features are uniformly distributed across space and scale.

To constrain the matching stage into a local search windows, an initial rotation of the image plane dealing with small motions in 3D is estimated. Feature matching stage associates the extracted features with descriptors, usually local binary patterns of normalized intensity values and features are matched using a mutual-consistency check. A robust estimate is performed by RANSAC [7] and the final transformation is estimated on the inliers. Robustness is further increased by "keyframes" that reduce drift when the camera viewpoint does not change significantly.

THE LASER PAINTBRUSH

Fig. 5 outlines the main process of laser pointer tracking. Since we use cameras with IR sensitive pixels, and emit IR from the laser, we can readily localize the laser pointer in the IR input images. In most cases, the IR images will contain high-intensity pixels associated with the laser dot and some spurious noise. Due to the presence of noise, we cannot simply extract pixels with the highest intensity. Hence we "track" a local window around the laser dot.

A user first initializes the tracker by pointing the laser into a predefined rectangle in the center of the image. To avoid the spurious noise (often with higher-intensity values than the laser dot), we need to keep the tracked window as local as possible. However, we also need to handle even rapid motion of the camera or the laser pointer itself (or both) resulting in a very large displacement in the image plane. To this end, the laser pointer tracker uses a Kalman filter, which predicts a pose of a local window in frame $t+1$. In this frame, we move the local window into the predicted position, and threshold the patch. The tracker automatically switches into the "re-detection" mode if the mean intensity of a patch is higher than some fraction α of the highest intensity, the thresholded pixels are not connected, or the number of thresholded pixels is much higher than the expected size of the laser dot at a given distance.

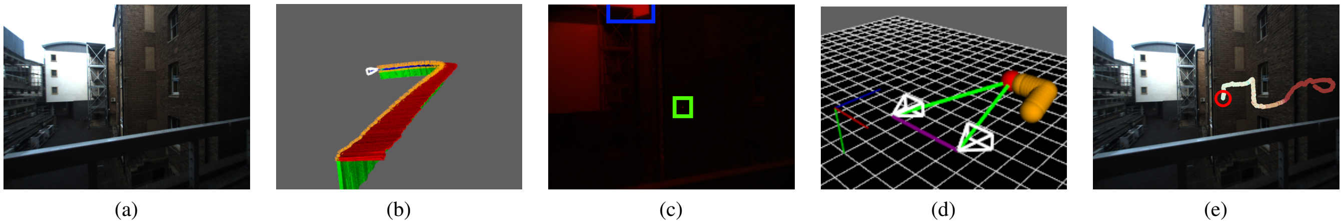

Figure 5: Laser pointer tracker: a) color images (left camera), b) part of visual odometry c) filtered images with local window (green) which is necessary since the image often contains brighter areas (blue), d) triangulation, e) tracked and triangulated (mapped) points projected onto the RGB frame.

Next, the measured pose is used to "correct" the Kalman filter prediction. If the laser tracker is in the "re-detection mode", it attempts to re-initialize in a small area close to the last known position – this is important since the laser dot intensity is often decreased to the level of noise (or can even completely disappear) due to diffraction and other optical effects. Further robustness is ensured by epipolar constraints; we run two laser trackers (left and right camera) in parallel and if the predictions do not satisfy the epipolar constraint, the tracker switches to the re-detection mode.

The final step is a 3D triangulation of the tracked 2D points observed in left and right images. To this end we implement simple and efficient linear triangulation as described in [13]. We then use the 6DoF pose estimated from the visual odometry as a means to back-project this 3D point into world space (from camera space). This adds a desirable temporal consistency of the 3D points, allowing 3D point tracks to be built up over time to detect 'brush' strokes and other gestures.

Interactive Improvement of Disparity

Whilst the laser pointer provides the main method for user input, it also carries another important benefit: The triangulated 3D point is very robustly matched and consequently a high-quality depth/disparity estimate is provided at a single sparse point. In general, dense depth/disparity estimation in large outdoor areas using passive stereo is a difficult problem. Although there has been a tremendous progress over the past decade, the typical output of a real-world sequence suffers from several issues – disparity estimates in homogeneous and/or over-exposed (saturated) areas are usually incorrect or completely missing. Moreover, most algorithms evaluate disparity independently per stereo-pair, *i.e.* disparity evaluated on a video sequence typically exhibits a "flickering" effect.

Our applications allows to, at least partially, recover incorrect and/or missing disparity values by laser interactions. A user labels the scene surface by laser pointer. The laser tracker is able to find such dots, which are triangulated and used to obtain high-quality depth estimates. These can be used as prior in disparity estimation algorithm. To this end, we modified approach of Geiger *et al.* [10]. A desirable side-effect is, that the tracked points are stable over time, hence also the prior is stabilized and the estimated disparity within regions corrected by interactions is more temporarily consistent.

Fig. 6 shows the steps used in disparity correction using the sparse 3D laser point. The method of Geiger *et al.* [10] robustly matches a set of sparse corner-like features (called support points) first and forms a Delaunay triangulation which serves as prior (a piece-wise linear function) in a generative model for stereo matching. This reduces the disparity search space to plausible regions and tends to disambiguate dense matching even without any global optimization method.

In order to improve disparity by interactivity, we inject the tracked laser points into a set of the support points before it computes triangulation. Feature matches in homogeneous areas are often incorrect, hence we form a convex hull around tracked points and remove all support points from its interior. We prefer this conservative strategy since we can always add missing matches by laser interactions; however, we need to make sure there are no incorrect matches (support points for prior). We consider three situations for each triangle: 1) if all three vertices of a triangle are laser points, we decrease the weight of feature matching term in the final energy function by β_{decr} (*cf.* [10], Eq. 8) and rely much more on the prior since we are very certain about the correct disparity value from the laser points. 2) if at least one vertex is a laser point, we decrease the weight of the feature matching term only by some fraction of β_{decr}. Finally, if no vertex of a triangle is obtained by laser pointer, we maintain the current state.

Unfortunately, all real-world measurements contain some level of noise, which includes our laser tracker, triangulation, and pose estimation (projection of 3D point to a current frame). In order to handle noisy input data and make disparity estimates more robust in the case of planar surfaces, we find connected components of triangles, fit a plane (using RANSAC with least squares refinement on inliers) into all support points obtained by the laser tracker, and share the estimated plane as a more robust prior (this can be viewed as a regularization) by all triangles within a connected component. In the case of unreliable surface measurements, a user can easily add more support points in these regions in order to clean them up during refinement.

Interactive Learning of Semantic Labels

In addition to map refinement, the laser is used to mark objects of interest through simple gestures such as strokes. Our system is then able to learn a classifier for labeled parts of a scene, where the semantic labels l_s are provided through speech recognition. Note that the user need not precisely label every voxel belonging to the object and can instead roughly mark a small set of voxel and the algorithm will automatically propagate labels to the other voxels belonging to the same part of a scene. Laser interaction is more convenient in this outdoor scenario than touch gestures, which are used in the SemanticPaint system [32]. The laser allows interaction at a distance and does not corrupt the 3D reconstruction; moving hands in front of a camera causes serious issues if corresponding depth values are not segmented and masked out properly.

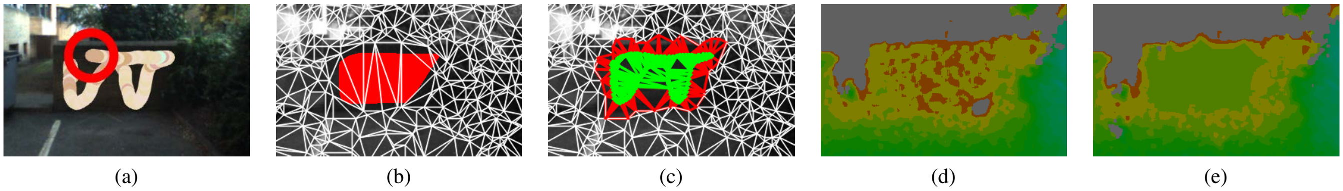

Figure 6: Disparity correction: a) tracked laser point, b) removed support points c) support points provided by interaction, d) disparity without interactivity, e) improved disparity

At the heart of our system is a densely-connected pairwise dynamic Conditional Random Field (CRF) defined on voxels. In such a model, each voxel i in the 3D reconstruction volume $\mathcal{V}$ of the scene is represented by a discrete random variable x_i that represents the semantic class $l \in \mathcal{L}$ (*e.g.* road, sidewalk, wall, . . .) that the voxel belongs to. The unary potentials are evaluated by streaming decision forests that are extremely fast both to test and train online on voxel oriented patches (VOP) features. For the pairwise potentials, we employ the standard Potts model that enforces smoothness but preserves edges (we use RGB, surface normal vector and 3D world coordinate as features). These potentials take form of mixture of Gaussian kernels that allow efficient, filter based inference which is a message passing algorithm. Although the CRF is defined on continuously changing data, its energy landscape changes only gradually from one frame to the next. This allow us to amortize the optimization cost over multiple frames and a GPU implementation allows super real-time speeds (one update of the messages requires 6 ms).

AUGMENTED VISUALIZATION

The last step of our pipeline involves rendering our synthetic scene on the (full-color) displays of the glasses. These displays are transparent, allowing our raycasted 3D model to be superimposed over the user's view of their physical environment. For interactive segmentation of the scene, superimposing the two in this way provides a natural way of interacting with the 3D model, providing users with a way to verify the accuracy of the interactive labeling of the scene in real time. Our rendering through the glasses shows various semantic classes using a number of easily-distinguished colors. Fig. 8 shows some visualization examples.

Interactive Reduction of Visual Clutter

In addition to its uses for semantic labeling and improving disparity, the laser pointer can also be used to select individual semantic classes for visualization. That is, the user points at a part of the scene that is labeled with a particular class and the system then highlights all parts of the scene that belong to that class, whilst graying out those parts that belong to other classes. This provides a useful way of reducing the visual clutter in a scene, *e.g.* it might be useful for a visually-impaired person trying to follow a footpath to be able to prominently highlight the footpath and grey out classes such as the road and the surrounding buildings. The user can either select a class that should be highlighted until the system is informed otherwise (which is useful for tasks such as following a footpath), or switch into interactive highlighting mode, in which case the class being highlighted changes in real time as the user moves the laser pointer around. This could also be augmented with audio feedback for visually impaired users to determine the type of objects in view.

Map Sharing

In order to allow optional multi-user interactions, we need to share information between users. For ease of exposition, we discuss a two user scenario. We assume that users A and B are close to each other so they observe almost the same part of the scene. In our scenario, only user A builds a single common map and provides raycasted visualizations to user B. At the beginning, we estimate a relative pose between the users and run visual odometry for each of them. Then, the user B sends only a 6DoF pose and receives a raycasted visualization from her own perspective so both users can interact. Though this approach adds computational load on the A, this is not an issue in practice, since raycasting takes only 5 ms. Though our map sharing method is simple and efficient, the quality depends on precision of visual odometry. In order to prevent drifting, we re-estimate a relative pose between the users every 500 frames.

EXPERIMENTAL RESULTS

Qualitative Results

Propagation of User Labels The user indicates the surface of objects in the physical world using the laser pointer. Our system interprets such indications as a paint stroke, and voice input is used to associate an object class label to the corresponding voxels. Then, our mean-field inference engine (see our related paper [32]) propagates these labels through the reconstructed scene very efficiently. Thanks to the pairwise potentials we use, the result is a spatially smooth segmentation that adheres to object boundaries. Examples of label propagation are shown in Fig. 7 and supplementary material.

Semantic Labelling Our system learns a streaming decision forest classifier in a background CPU thread given the labels provided by the user. At some point, the user selects 'test mode', and the forest starts classifying all voxels. In Fig. 8 and supplementary material shows the final smooth results obtained by running our mean-field inference procedure on our decision forest predictions. We show results on four new challenging sequences captured using a head-mounted display (the last two columns belong to the same sequence). The images clearly indicate the sharp boundaries that we manage to achieve between different conflicting semantic classes. For example, observe the extremely accurate boundary between the pavement and the road in the sequence in the third column.

Quantitative Results

We evaluate the accuracy of our mean-field filtering of the forest predictions, based on a variety of test sequences captured from outdoor scenes. For each sequence, a series of keyframes were hand-labeled with object segmentations. Keyframes were selected to ensure full coverage of the scene. These ground-truth images are then projected and aggregated onto the underlying TSDF, and then back-projected to all the views of each sequence. We use the results to calculate global accuracies for each of our semantic classes (see Table 1).

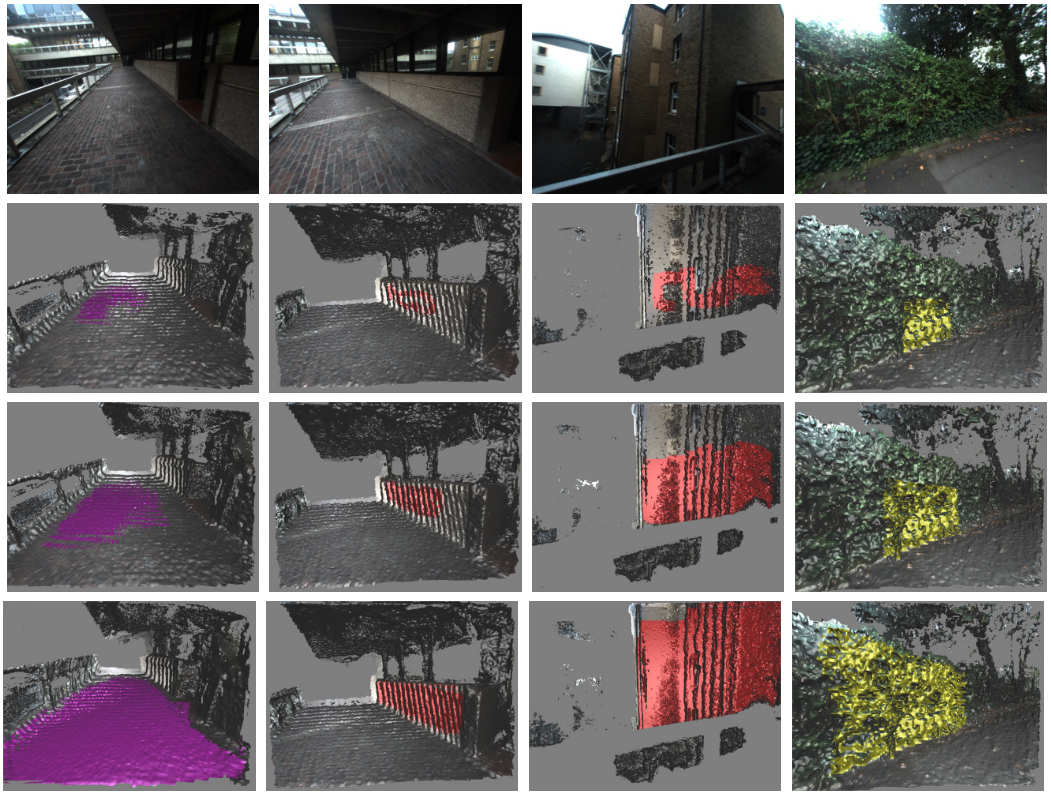

Figure 7: Label propagation. Our efficient inference engine smoothly propagates class labels from the voxels indicated by the user to the rest of the volume. Here we show examples from four sequences. The first row shows the raw sequences; the second row shows the labelling after a couple of propagation steps; the third and fourth rows shows the labelling at later stages of the propagation. The pairwise terms in our energy encourage a smooth segmentation that respects object boundaries.

Class	Bgd.	Bld.	Road	Pavement	Tree	Bin
2D Labelling	87.6	85.4	86.2	86.9	82.3	80.5
3D Labelling	88.5	89.3	88.9	89.2	89.3	84.0

Table 1: Global accuracies (true positives / total numbers) for each class we use. The first column shows results for labelling in the image domain; the second column shows results for labelling in 3D and then projecting those labels to 2D.

Computational Efficiency

The inherently volumetric nature of our approach parallelizes well on modern GPU architectures. For our experiments we employed laptops with an Nvidia GeForce GTX 880M with 8 GB of GPU RAM, and quadcore Intel i7 processor with 24 GB of CPU RAM. We provide approximate system timings in Table 2. Although the timings change as a function of the number of visible voxels and resolution, in all tests we observed interactive frame rates. Numbers are provided for $4 \times 4 \times 4$ cm^3 voxel resolution, 1024×768 pixels and we do not fuse voxels beyond 20 m. The semantic segmentation pipeline runs on a GPU while the disparity estimation, laser tracker, disparity correction, visual odometry and forest learning run on the CPU. Note that the forest learning runs asynchronously in a background thread so it does not influence reconstruction and labeling. This thread continuously samples new labeled training data from the current view frustum and updates itself. This ensures an up-to-date forest is available for classification whenever the user requests it. As shown in Table 2, the most time consuming step is disparity estimation, but this can be implemented on GPU as well. Note, the reconstruction and labeling are independent of forest update step.

For a two-user scenario, we established a peer-to-peer wireless network. Since we transfer only 6DoF pose and raycasted visualizations, the data transfer is fast enough. Latency is not a huge issue, since we accumulate all the interactions over multiple frames and transfer the data when user is satisfied.

Finally, the size of the environment that our system is able to map is limited by 1) drift of the visual odometry, 2) battery life of a laptop under heavy load (1 hour) and 3) GPU and CPU RAM. Considering these limits, we were able to run our system in environments of up to 100-500 meters. A standard scene can be reconstructed and labeled at human walking speeds and we use no post-processing.

APPLICATIONS

So far we have described and evaluated our novel mapping system, and uncovered its low-level interactive capabilities. Whilst the focus of our work is technical, we believe the system as a whole could have dramatic impact for HCI applications. We demonstrate one potential application area in the next section, and discuss others afterwards.

Semantic Maps for the Visually Impaired

There are more than 285 million people in the world living with sight loss which has a significant impact on their daily lives. Over 85% of these individuals have some remaining vision [20]. Recently, there has been an interest in developing smart glasses [14, 8], which seek to provide these people with additional information from the nearby environment through stimulation of the residual vision. The aim is to increase the information level regarding the close environment using depth and/or image edges. This rather simplistic, though effective,

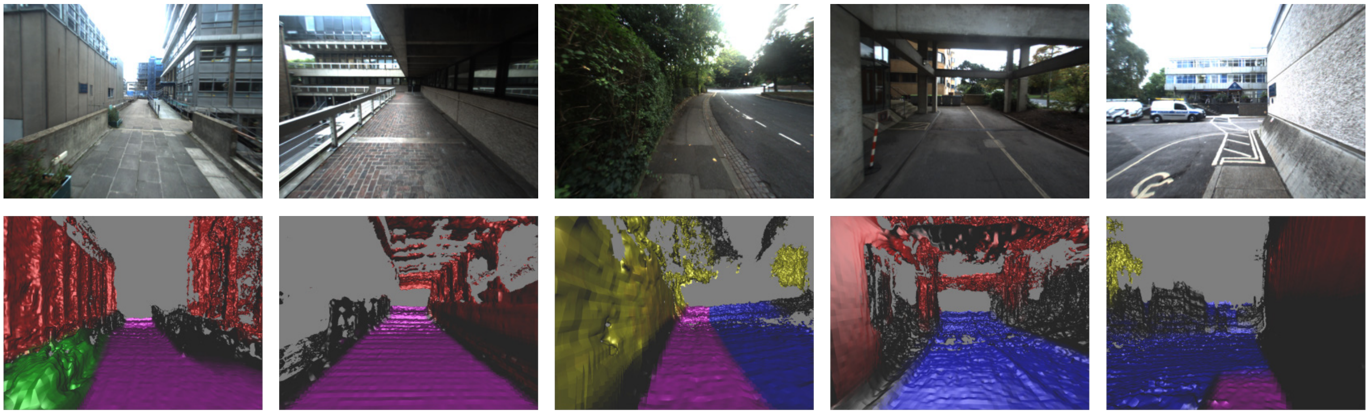

Figure 8: Final mean-field inference results for four sequences (the last two columns belong to the same sequence). Our streaming decision forest is able to learn to make per-voxel predictions about the object classes present in the scene. Each pixel is classified independently, and so the forest predictions can be somewhat noisy. The mean-field inference effectively smooths these predictions to produce a final labelling output to display to the user.

Disparity Estimation	Laser Tracker	Disparity Correction	VO	Fusion	Forest Update	Forest Evaluation	Mean-field	Wifi latency
80 ms	2 ms	3 ms	20 ms	15 ms	170 ms	5 ms	2-10 ms	5-10 ms

Table 2: Approximate system timing per frame. Despite small fluctuations we observed consistently good, interactive frame rates. Note, the reconstruction and labeling are independent of forest update step.

method for information extracting, enables the user to more independently traverse and navigate areas by providing the user with richer information than residual vision could provide.

We believe our live semantic maps can be directly used to highlight user-specific objects learned through online teaching with a carer/helper/trainer. This will present the visually impaired user with even more information regarding the nearby environment to understand the surrounding environment more clearly. Examples of these user-specific objects or regions could be stairs, road-crossings, bus-stops, entry or exit doors, sidewalk, restrooms, ticket machines or booths.

The basic scenario is that a visually impaired and a helper, both wearing smart-glasses displaying individual views of a shared 3D reconstructed environment, label the user-specific objects or regions of interest through usage of a laser pointer handled by the helper. The user can learn to use the system within familiar environments highlighting only regions that the user finds useful with help from the carer. The objects are labeled and learned by the system online, hence immediate response can be provided to both users. Fig. 9 shows a demonstration of this scenario.

At a later stage, once the objects are labeled, the visually impaired user can return to the same scene, and view the semantic map using the heads-up display. The transparency of the displays is advantageous for a visually-impaired person using the glasses to navigate around a pre-labeled scene, since it allows them to enhance, rather than replace, their remaining vision with the spatial information provided by the 3D mapping. Furthermore, transparent displays allow other people to see the wearer's eyes, which is helpful for social interaction.

Once a scene has been labeled by a sighted user, it can easily be converted into a form that is suitable for assisting a visually-impaired person to understand the nature of their environment, and navigate safely around it. Existing techniques such as those of Hicks *et al.* [14] have shown the usefulness of whole-image techniques, such as depth-to-brightness mapping, for helping visually-impaired individuals to avoid obstacles. The inclusion of semantic labeling has the potential to add an additional dimension to this kind of system by providing the wearer with more information about the objects around them and the boundaries between surfaces in their local environment (*e.g.* between a footpath and a road).

Other Applications for Semantic Maps

Personalized semantic maps with known object segmentations could also be used for a variety of other way-finding and navigation applications, either for robots or end users. For example, imagine self driving cars or quadcopters being able to follow particular paths and avoid obstacles. Additionally, users could interact with these robots, asking them to find particular instances of objects by semantically breaking down the world and using the laser pointer (*e.g.* 'please go to *that* building'). Furthermore, if such a model was maintained and updated over time, finding points of interest could be as simple as uttering a few words (*e.g.* 'where is the nearest bus stop').

These personalized maps could also be interactively captured, shared online and played back. For example, a user could give fine details for navigation to a friend, by actually capturing their path through the city, and then sharing it online, allowing for a detailed retracing of the steps, potentially with audio feedback. Another aspect is the ability for users to add semantic information to online maps. Here, by crowd-sourcing multiple personalized maps, a larger corpus of semantic maps could be generated. Users could use these semantic labels for searching, *e.g.* 'find the nearest bus stop to my map location', or 'please find the entrance to the building'. This latter point is also very important, as it allows a level of detail not yet available in regular maps, allowing for a more fine-grained level of way finding and navigation. Finally, augmented reality mobile gaming could be a rich source of application. Imagine quickly scanning in and labeling an outdoor space, and then associating object classes with aspects of the game. For example, game characters could hid behind particular objects, or follow particular paths or enter buildings. Such augmented reality scenarios could be expanded for planning the renovation of buildings and cities, automating inventory, and town planning.

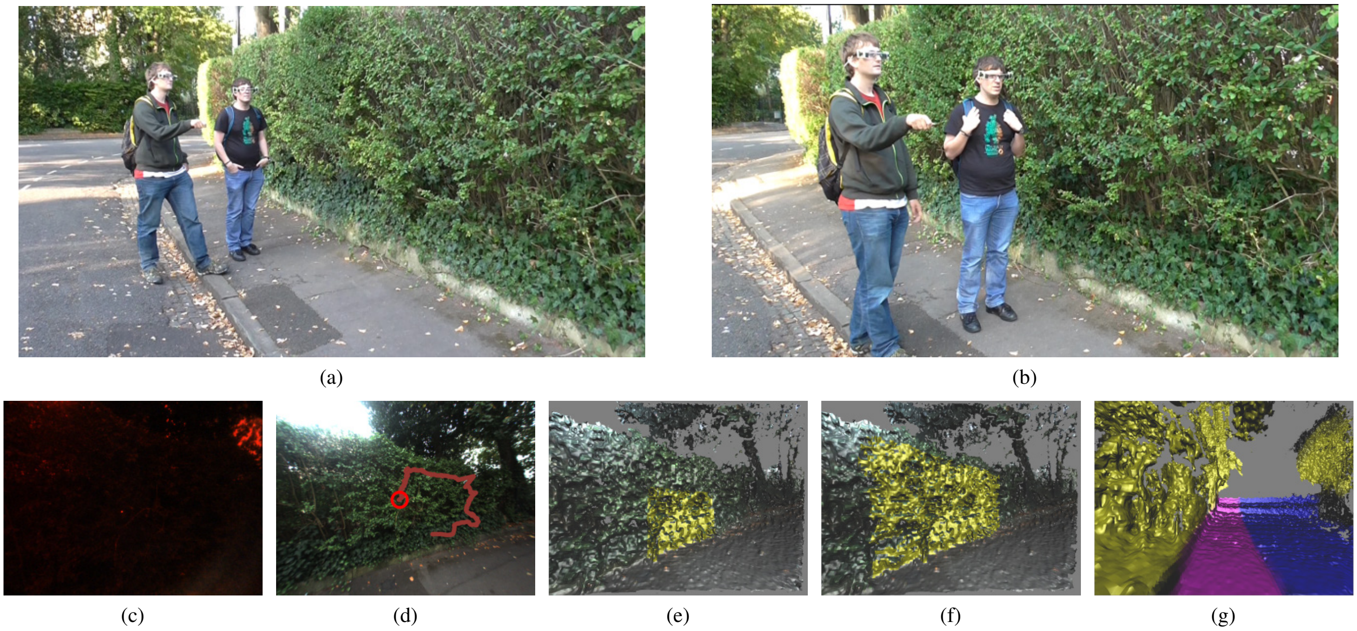
(a) (b) (c) (d) (e) (f) (g)

Figure 9: A potential application area for semantic 3D maps for aiding visually impaired people to navigate outdoor spaces. The basic scenario is that a visually impaired and a helper, both wearing smart-glasses displaying individual views of a shared 3D reconstructed environment, label the user-specific objects or regions of interest through usage of a laser pointer handled by the helper. As shown in (a and b) the helper is indicating the object 'tree' using the laser pointer. Our system then starts to learn this 'tree' model. We first track and detect the laser dots in the IR images (c). These points are detected and tracked over a sequence of frames (d). After interaction, the label propagates to segment the tree (e and f), and other instances are detected in the scene (g).

User Experience The proposed system has been used by 15 users. All users felt comfortable with it. In particular, they liked the system performing a 3D reconstruction and labeling at interactive framerates and the laser pointer providing a natural means of interacting at a distance in outdoors environment as opposed to touching. The users also liked the see-through glasses allowing to see the real world with overlayed outputs providing an extra information about the environment.

Though the users provided a positive feedback in general, they also suggested a few modifications to make the system more comfortable, mostly on the hardware side. They suggested in particular to balance the center of gravity of the AR glasses better to prevent sliding off from the user's nose. Another recommendation was to change the position of wires in order to less resctrict the motion of user's head. On software side, the users mostly complained about drift of the visual odometry.

LIMITATIONS Despite very encouraging results, our system is not without limitations. As with all recognition algorithms, the segmentation results are not always voxel-perfect, as shown in the results and accompanying video. One possibility, however, is to allow the user to interactively make corrections to help reduce such errors. We believe additional modes of interaction such as voice priors (*e.g.* 'walls are vertical'), as well as more intelligently sampling the training examples could further improve results. From a computational standpoint, our system is fairly GPU heavy, which limits us to laptop only uses currently. With the advent of mobile GPGPU there are likely ways of addressing this in future work.

Further, our system is currently state based; *i.e.*, it requires the use of voice commands to switch between annotation, training, and test modes. We are planning an extension where both the learning and forest predictions are always turned on. This will require considerable care to avoid 'drift' in the learned category models: the feedback loop would mean that small errors could quickly get amplified. Finally, algorithmic parameters such as the pairwise weights are currently set at compile time (these are cross-validated and common across datasets shown). Given a small training set (perhaps boot-strapping), more reliable settings could be automatically selected online.

Since our system uses an IR laser pointer with output far less than 5mW, the pointer will not work in direct sunlight, but the IR laser can be replaced *e.g.* by LIDAR-based pointer. In general, the system works well in an urban environment, but fails in areas where the visual odometry and/or disparity estimation fail (*e.g.* those containing highly reflective or specular surfaces, or textureless regions).

CONCLUSIONS

In this paper, we have presented an interactive 3D mapping system that can semantically label large unknown outdoor scenes. The system can take advantage of interactive input from the user in order to guide the mapping towards objects and elements of interest in the scene. Rather than using active depth cameras, we capture our input using a passive stereo approach, making it possible to reconstruct large or distant structures outdoors.

Our system comprises a pair of see-through glasses, two RGB-Infrared stereo cameras, and a one-button laser pointer. The laser pointer helps the user highlight objects of interest and, in combination with voice commands, can provide semantic labels for objects (even distant objects) in an online fashion. The laser pointer can also be used to provide accurate, sparse measurements to the system in order to improve the estimated stereo depth, and thereby improve the final reconstruction.

We believe our mapping system could be of particular use for the visually-impaired, who in many cases can benefit from a more accurate understanding of the nature of objects in their environment. Whilst we have focused our work on the technical details, we feel this could be a high impact area for future work. For example, our system's ability to differentiate footpaths from roads has the potential to be extremely helpful in providing visually-impaired people with a safer way to navigate independently outdoors.

REFERENCES

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R. Building Rome in a Day. *CACM* (2011).
2. Chen, D. M., Baatz, G., Köser, K., Tsai, S. S., Vedantham, R., Pylvänäinen, T., Roimela, K., Chen, X., Bach, J., Pollefeys, M., Girod, B., and Grzeszczuk, R. City-scale landmark identification on mobile devices. In *CVPR* (2011), 737–744.
3. Curless, B., and Levoy, M. A volumetric method for building complex models from range images. In *SIGGRAPH* (1996), 303–312.
4. Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. MonoSLAM: Real-Time Single Camera SLAM. *PAMI 29*, 6 (2007).
5. Engel, J., Schöps, T., and Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *ECCV* (2014).
6. Engel, J., Sturm, J., and Cremers, D. Semi-Dense Visual Odometry for a Monocular Camera. In *ICCV* (2013).
7. Fischler, M. A., and Bolles, R. C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *CACM 24*, 6 (1981).
8. Froissard, B., Konik, H., Trmeau, A., and Dinet, . Contribution of augmented reality solutions to assist visually impaired people in their mobility. In *Universal Access in Human-Computer Interaction. Design for All and Accessibility Practice*. Springer, 2014, 182–191.
9. Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. Reconstructing Building Interiors from Images. In *ICCV* (2009).
10. Geiger, A., Ziegler, J., and Stiller, C. StereoScan: Dense 3d Reconstruction in Real-time. In *IVS* (2011).
11. Habbecke, M., and Kobbelt, L. LaserBrush: A Flexible Device for 3D Reconstruction of Indoor Scenes. In *SPM* (2008).
12. Hane, C., Zach, C., Cohen, A., Angst, R., and Pollefeys, M. Joint 3d scene reconstruction and class segmentation. In *CVPR* (2013), 97–104.
13. Hartley, R., and Zisserman, A. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
14. Hicks, S. L., Wilson, I., van Rheede, J. J., MacLaren, R. E., Downes, S. M., and Kennard, C. Improved mobility with depth-based residual vision glasses. *Investigative Ophthalmology & Visual Science 55*, 5 (2014).
15. Huang, A. S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., and Roy, N. Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera. In *ISRR* (2011).
16. Iannacci, F., Turnquist, E., Avrahami, D., and Patel, S. N. The Haptic Laser: Multi-Sensation Tactile Feedback for At-a-Distance Physical Space Perception and Interaction. In *CHI* (2011).
17. Jr., D. R. O., and Nielsen, T. Laser Pointer Interaction. In *CHI* (2001).
18. Klein, G., and Murray, D. W. Parallel tracking and mapping for small ar workspaces. In *ISMAR* (2007).
19. Ladický, L., Russell, C., Kohli, P., and Torr, P. H. S. Associative Hierarchical CRFs for Object Class Image Segmentation. In *ICCV* (2009).
20. Mariotti, S. P. Global Data on Visual Impairments 2010. Tech. rep., World Health Organization, 2010.
21. Munoz, D., Bagnell, J. A., and Hebert, M. Stacked Hierarchical Labeling. In *ECCV* (2010).
22. Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *ISMAR* (2011).
23. Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. DTAM: Dense Tracking and Mapping in Real-Time. In *ICCV* (2011).
24. Nguyen, T., Grasset, R., Schmalstieg, D., and Reitmayr, G. Interactive syntactic modeling with a single-point laser range finder and camera. In *ISMAR* (2013).
25. Nießner, M., Zollhöfer, M., Izadi, S., and Stamminger, M. Real-time 3d reconstruction at scale using voxel hashing. *TOG 32*, 6 (2013), 169.
26. Qin, Y., Shi, Y., Jiang, H., and Yu, C. Structured Laser Pointer: Enabling Wrist-Rolling Movements as a New Interactive Dimension. In *AVI* (2010).
27. Rosten, E., and Drummond, T. Machine learning for high-speed corner detection. In *ECCV* (2006).
28. Salas-Moreno, R. F., Newcombe, R. A., Strasdat, H., Kelly, P. H. J., and Davison, A. J. SLAM++: SLAM at the Level of Objects. In *CVPR* (2013).
29. Sengupta, S., Greveson, E., Shahrokni, A., and Torr, P. H. S. Urban 3d semantic modelling using stereo vision. In *ICRA* (2013), 580–585.
30. Taneja, A., Ballan, L., and Pollefeys, M. City-scale change detection in cadastral 3d models using images. In *CVPR* (2013), 113–120.
31. Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. Bundle adjustment - a modern synthesis. In *Workshop on Vision Algorithms* (1999).
32. Valentin, J., Vineet, V., Cheng, M.-M., Kim, D., Shotton, J., Kohli, P., Niessner, M., Criminisi, A., Izadi, S., and Torr, P. H. S. SemanticPaint: Interactive 3D Labeling and Learning at your Fingertips. *ACM TOG* (2015).
33. Valentin, J. P. C., Sengupta, S., Warrell, J., Shahrokni, A., and Torr, P. H. S. Mesh based semantic modelling for indoor and outdoor scenes. In *CVPR* (2013), 2067–2074.
34. Whelan, T., Johannsson, H., Kaess, M., Leonard, J. J., and Mcdonald, J. Robust real-time visual odometry for dense rgb-d mapping. In *ICRA* (2013).
35. Wienss, C., Nikitin, I., Goebbels, G., Troche, K., Göbel, M., Nikitina, L., and Müller, S. Sceptre – An Infrared Laser Tracking System for Virtual Environments. In *VRST* (2006).
36. Xiong, X., Munoz, D., Bagnell, J. A., and Hebert, M. 3-D Scene Analysis via Sequenced Predictions over Points and Regions. In *ICRA* (2011).

This research was supported by Technicolor, EPSRC, the Leverhulme Trust, the ERC grant ERC-2012-AdG 321162-HELIOS, the NIHR i4i grant II-LB-1111-20005 and a Royal Society Brian Mercer Award for Innovation.