
Link to published version (if available): http://dx.doi.org/10.1109/ICRA.2013.6630877

Link to publication record in Explore Bristol Research
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: http://www.bristol.ac.uk/pure/about/ebr-terms
Visual mapping using learned structural priors

Osian Haines, José Martínez-Carranza and Andrew Calway
Department of Computer Science
University of Bristol, UK

Abstract—We investigate a new approach to vision based mapping, in which single image structure recognition is used to derive strong priors for initialisation of higher-level primitives in the map. This can reduce state size and speed up the building of more meaningful maps. We focus on plane mapping and use a recognition algorithm to detect and estimate the 3D orientation of planar structures in key frames which are then used as priors for initialising planes in the map. The recognition algorithm learns the relationship between such structure and appearance from training examples offline. We demonstrate the approach in the context of an EKF based visual odometry system. Preliminary results of experiments in urban environments show that the system is able to build large maps with significant planar structure at average frames rates of around 60 fps whilst maintaining good trajectory estimation. The results suggest that the approach has considerable potential.

I. INTRODUCTION

In vision based mapping, whether for odometry or simultaneous localisation and mapping (SLAM), early and fast instantiation of 3D features improves performance – not only in terms of robustness and stability of pose tracking, but also in providing structural information as soon as it becomes visible. The latter is especially important in real-time applications involving navigation and interaction with the environment. For example, in stereo, and more recently RGB-D camera systems, point features are initialised with strong depth priors, giving faster and improved map building, see e.g. [1] and [2]. Also, careful selection of feature combinations for initialisation can yield faster convergence of 3D estimates and hence better mapping and localisation as described in [3].

In this paper we introduce a new approach to speeding up map building, motivated by the following observation. If we have knowledge of the geometry of entities such as objects or structural primitives, and a means of detecting their presence in a single frame, then it provides a quick way of deriving strong priors for constructing the relevant portions of the map. In the extreme, we might envisage instantaneous insertion of known objects and structural elements, derived, for example, from scene specific CAD models, thus reducing map building to model alignment. Our interest, however, is in the more general middle ground: can we use knowledge of the appearance and geometry of primitive classes, such as buildings, roads, trees, etc, to allow fast derivation of strong priors for directing feature initialisation?

To illustrate, we focus specifically on map building with planar structure, an extension of point based mapping which has received attention due to the ubiquity of planes in urban and indoor environments. Approaches include fitting planes to point clouds [4], use of Manhattan models [5] and growing planes alongside points [6], [7]. Although these methods have shown the potential advantages this can bring, they are also handicapped by having to allow sufficient parallax (and hence time) for detecting planes in 3D, whether it be by fitting or growing. This is a good example of where the early availability of priors would be beneficial. Indeed, this was nicely demonstrated in the work of Castle et al. [8], in which specific planar objects with known geometry were detected and inserted in the map, giving improved tracking and fast generation of a rich map representation.

We seek to generalise this, aiming to derive strong priors for the location of planar structure per se, without reference to specific planes. For this we use a machine learning algorithm which is able to recognise planes in single images, based on learning the relationship between appearance and structure from training examples [9]. The algorithm detects planar regions in images and gives an estimate of their surface normal. Thus, for a given frame, as illustrated in Fig. 1, this provides a prior for both the projected location and orientation of likely planar structure which we can use to initialise planes in the map. We demonstrate this in the context of an extended Kalman filter (EKF) visual odometry (VO) system, modifying our plane growing method previously described in [7].

In the next section we provide an overview of the system, followed by details of each component in Sections III and IV. In Section V we present results of experiments in urban environments. These show that the approach is capable of
incorporating larger planar structures into the map and at a faster rate than previously reported in [7] – averaging around 60 fps – while still giving good pose trajectory estimates. This demonstrates the potential of the approach both for the specific case of planar mapping and the wider aspect of using image recognition techniques to generate useful priors for map building.

II. OVERVIEW

Our combined plane detection and visual odometry (PDVO) method is based on our VO system described in [7], which recovers the trajectory of a monocular camera by building a map incorporating point and plane features. This builds planes over multiple frames by iteratively growing from a set of initial seed points, so as to find local clusters obeying a planar constraint. This can successfully find planes when available, and falls back to using point features otherwise, courtesy of a common feature representation – the inverse depth planar parameterization (IDPP). The drawback is that many seed points must be initialised and grown in order to find those which actually belong to planar structures; plus there is always the risk of introducing planes in inappropriate areas, especially when the features are distant, or the camera is performing pure rotation. The search is performed blind, with no prior knowledge of where the planes are: this is where a learned structural prior becomes useful.

The base VO system is modified to use prior information by using a plane detection method (see section III) to identify planes in a single frame – this gives an area of the image, and a 3D orientation estimate, for each detected plane. When the VO system decides that more features are needed, the plane detector is called. The result is then used to initialise planes in the map and to direct point feature initialisation on those planes in order to refine their depth and orientation. This means that arbitrary seed point growing is avoided, ensuring that costly measurements are targeted towards likely planar structure, gaining improved mapping and localisation. The result is that we quickly build a map consisting only of planes corresponding to semantically correct regions; Fig. 1 illustrates the main steps of the method.

We emphasise that this is a significantly different approach from other plane-based VO or SLAM systems; in the past planes are either detected after the map is built [4] or concurrently [7], whereas we use the appearance of a single image to directly find planar structures before they are mapped, and use this information to guide feature initialisation. As well as being potentially much faster than methods which need multiple images, this is exploiting a different type of information – the cues available in a single image – which are normally ignored in purely geometry-driven systems.

III. PLANE DETECTION

Planes are detected using the method we introduced in [9]: this detects planes, and estimates their orientation, from only a single image – i.e. without using cues such as depth or optical flow. In contrast to most methods for single-image perception, it does not rely on specific geometric cues (such as vanishing lines or characteristic texture distortion), and instead learns directly from example images to predict class and orientation, by using machine learning techniques; this makes it applicable to a wider variety of scenes, and so is appropriate for our application in general outdoor environments.

To find planes, the method uses a plane recognition algorithm [10] which classifies, for a given image region, whether it is a plane or not, and if so, estimates its orientation. This process is illustrated in Fig. 3: given a region of the image, salient points are detected, and the patch surrounding each is described using a histograms of gradients, and a RGB colour histogram. To compactly represent the whole region, these descriptors are quantised against a pre-built bag of visual words, to form a histogram of word occurrence (one each for gradient and colour words). These are combined using a variant of latent topic analysis [11], which also serves as dimensionality reduction, to represent the region as the occurrence of latent topics (each being a weighted combination of words). To represent the spatial distribution of topics, spatiogram descriptors are used [12]: these represent not only the occurrence of each topic, but also the mean and covariance of the points contributing to each topic. Regions are classified, using the spatiogram descriptors, with a Relevance Vector Machine (RVM) [13]. The RVM is trained using spatiograms generated for a large set of manually annotated training data, each labelled with their class and orientation.

The above can effectively determine whether regions are planar, and predict their orientation, but of course initially the location and extent of potential planes in an image is not known. Plane detection is achieved by repeatedly applying the recognition algorithm to overlapping windows across the image (Fig. 2b), to find the most likely locations of planes. A robust estimate of the planarity and orientation at each salient point in the image is calculated from the classifications of all the windows in which it lies, producing the ‘local plane estimate’, as shown in Fig. 2c – in which the underlying structure of the scene is apparent. To extract planes from
this, a graph over the points is segmented using a sequence of Markov random fields (Fig. 2d) – first to separate planes from non-planes, then to segregate planes from each other using their estimated orientations. The final step is to apply plane recognition once more to the planar segments, to re-estimate their orientation. Further details of the algorithm and quantitative results can be found in [9]; some examples of detections while running alongside visual odometry are shown in Fig. 5, which show the location and extent of detected planes on the image, with neighbouring but non-coplanar surfaces being separated and assigned individual orientations.

IV. VISUAL ODOMETRY USING PLANE PRIORS

In this section we set out the key elements of the PDVO system. As indicated earlier, it is based on our VO system described in [7]. This uses an EKF framework, operating as in a standard visual SLAM engine, but with the modification that features that move out of view are removed from the state. Two feature types are used, single points and planes, and these are extracted using the IDPP representation which allows planes to evolve from points as and when appropriate. Planes are defined in terms of collections of point features which satisfy a planarity constraint and planes are measured via the projection errors of their point features. Full details can be found in [7]. In the following we point out the key components relevant to the development of the PDVO system.

A. Keyframes

An important feature of the VO system is the use of keyframes. These are used to relate the initial camera image of a planar feature with a reference camera stored in the state, which allows the current measurements of features to be related to the original view; measurements of points are made by warping their image, using the plane parameters, to match against the keyframe. If points are consistently matched they are retained in the plane, and growing moves outward; otherwise, they are discarded. Not only is the warping-based image match faster than descriptor-based matching [14], but helps to discard non-coplanar points: if a point is not on the same plane, its warped image will become less similar as the viewing angle changes, so it will not be measured, and eventually discarded. This keyframe-based representation is very convenient when performing plane detection, as will become apparent.

B. Adding structural priors

We use the plane detection algorithm with the visual odometry as follows. When it is decided that more features need to be initialised (generally when the number of measurements falls below a threshold) a new keyframe is inserted. Instead of initialising many seed points and attempting to grow each into a plane, seed points are only initialised at the centroids of detected planes. Crucially, the whole plane is not initialised immediately: while we have a good prior that there is a plane region, measuring the whole region as a plane could lead to errors, if some outliers are also included. Therefore the same region growing method is employed, but allowed to proceed at a faster rate, since it can be more confident that the surroundings are planar. Points which are not actually part of the same planar surface are automatically pruned by the algorithm’s 3D consistency test, so minor errors in the plane detection stage do not cause problems in the map estimation.

The other important piece of information provided by the plane detector is the orientation of the plane. This is used to initialise the normal in the filter, and while some error is expected in the value, it will likely be closer to the true value than an uninformed default value (planes were previously assumed to face toward the camera), so will allow faster convergence of the normal. This will also help ensure non-planar points are not used for measurements, since as described above, their warped image will be less similar if the correct normal is used. This, combined with the computational savings achieved by limiting plane growing to the detected regions, allows for a substantial increase in frame rate, which is important in time critical applications.

C. Execution time

While the plane detector requires only a single frame, it is not fast enough to run before the next frame is available – currently it runs at around 0.7 seconds per image. This is not a problem, since the detection can run in the background, in a separate processor thread, and the result is used once it has finished. The keyframe-based nature of the plane mapping is ideal in this respect, in that the planes can be added directly to the keyframe in which they were detected, then measured from the current frame, rather than attempting to reconcile the plane-detected image with the current camera view. We found that this delay did not introduce any problems, and was able to increase the speed of plane acquisition compared to the
undelayed growing method of IDPP (see Fig. 4 in our results section).

It could be argued that since it takes the duration of multiple frames to detect planes from one still image, it would instead be desirable to use all of these frames in standard multi-view plane detection approaches [4], [15]. However, all such methods depend strongly on the baseline – which is a serious problem if the camera is not moving or observing distant planes. On the other hand, this plane detection method, by using a single frame, will work even when the camera is motionless; and is able to exploit a different type of information (i.e. appearance) compared to the geometric plane growing – and so has a distinct advantage over standard alternatives.

D. Scene representation

To quickly build a planar map of the environment, we retain the plane estimates provided by the visual odometry, even when they are no longer being observed (that is, they have been removed from the filter and do not contribute to the map estimate); since the visual odometry combined with plane detection can quickly give fairly good estimates of planar structures, fixing them in this manner is sufficient to give a coarse scene representation. Although one consequence of this technique is that any planes which have not fully converged while in view are fixed in the map with incorrect measurements, this does not compromise the accuracy of the rest of the map.

We note that since visual odometry does not update the whole map as more observations are made, the accumulation of errors will inevitably lead to drift, in position and scale – this is a well known problem with visual odometry, especially monocular visual odometry when the absolute scale is not observable; existing methods to reduce drift could be used in conjunction with our method [16], but that is not the focus of our current work.

V. Experimental results

A number of experiments were carried out using videos of outdoor urban scenes. These were recorded using a handheld webcam running at 30Hz, of size $320 \times 240$ pixels and undistorted to correct for distortion introduced by a wide-angle lens. In these experiments, our primary aim was to investigate what is possible when using learned priors, rather than to run a comprehensive evaluation against the IDPP method. As such, we tuned both methods to perform well – in terms of accuracy and runtime – by observing how they fared as various parameters were changed (such as the number of measurements made per frame and the radius in which to search for new points). However, while this means we cannot definitively state the performance difference due to PDVO when all other parameters are held constant, it is important to note that using plane priors gives us more freedom to use different settings – an important example being the rate at which planes may grow, as discussed in section IV-B; and we found PDVO to be generally more robust to the choice of parameter values than IDPP.

First we consider the implications of the delay in initialisation while waiting for the plane detector to run, as explained in section IV, compared to the undelayed initialisation of (seeds of) planes by IDPP. In Fig. 4 we show the development of a keyframe over several frames after initialisation in both methods. The first row shows the initial input image, and the result of plane detection used to initialise planes in the PDVO system. Following this are images showing the progression of plane estimation; it is clear that IDPP, in the left column, quickly initialises many planes, at many image locations (some of which are not at all planar); these take some time to grow, and compete for measurements. When using plane detection, however (right column), a single plane is initialised at the centroid of the detected region, and allowed to grow quickly: even though there was a delay of around 14 frames before the detector finished, the resulting plane expands rapidly, overtaking those initialised by IDPP in number of measurements and image coverage. The bottom row shows 3D visualisations of the planes (this is their status corresponding to the last camera image shown); the many planes created by IDPP have not yet attained good poses, while the plane initialised in PDVO already shows appropriate orientation. Again, this difference
only occurs because the plane prior allows us to choose a less conservative rate for plane growing, highlighting that fact that the two methods operate in fundamentally different ways.

Next, we compare the two methods on a long video sequence, as the camera traverses a large loop of approximately 300 metres – this is a square surrounded by houses, with trees on the inside (the Berkeley Square sequence). 3D views resulting from the two methods are compared in Fig. 6; while both have recovered an approximately correct trajectory (note that the true path is not actually square, but the ends should meet) and have placed planes parallel to the route along its length, it is clear in the PDVO method (right) that there are fewer planes, which tend to be larger and less cluttered, giving a clearer representation of the 3D environment. Oblique views on the right show this clearly – compared to PDVO, the planes mapped by IDPP are smaller, more irregular, and with more varying orientations. We also show results for another video sequence, taken in an urban environment surrounded by planes on all sides (the Denmark Street sequence), shown in Fig. 7. Again, the map created using our method is more complete and clear than that with the original IDPP, with fewer and larger planes. More mapping examples, as seen from the camera, are shown in Fig. 10, and further examples of plane detections acquired during mapping are shown in Fig. 5.

Table I compares statistics calculated from mapping the Berkeley Square sequence, in order to quantify the apparent reduction in clutter. These confirm our intuition that when using plane priors, fewer planes will be initialised, by avoiding non-planar regions. Furthermore, the planes resulting from PDVO are measurably larger, both in terms of the average number of point measurements, and number of pixels covered.

As we emphasised earlier, our intention is to show the potential for using the plane detection method for fast map building, and not necessarily to produce a more accurate visual odometry. However, it is interesting to analyse the accuracy of PDVO compared to IDPP against the areas’ actual geography. The ground truth was not available for the sequences we use, but the trajectories can be manually aligned with and overlaid on a map – as shown in Fig. 8 for the Berkeley Square sequence, and for the Denmark Street sequence in Fig. 9. The latter is a compelling example, and suggests that, under certain conditions, our method helps to ameliorate the problem of scale drift (a well known problem for monocular visual odometry [17]); of course, many more repeated runs would be needed to quantify this, but we consider these initial tests to be good grounds for further investigation.

One of our main hypotheses was that by using strong structural priors, we can make mapping faster by more carefully selecting where to initialise planes. Our experimental results confirm this, shown in Fig. 11, where we compare the computation time (measured in frames per second) for both methods, on the Berkeley Square sequence. As previously reported in [7], the IDPP system achieves a frame rate of between 18 and 23 fps (itself an improvement on similar methods running at 1 fps [7]), which is confirmed by this experiment (blue curve). Our method clearly out-performs this, achieving both a substantially higher average frame rate of 60 fps and being consistently faster throughout the sequence, even though it is simultaneously running the plane detector. We are not aware of existing visual odometry systems running at such high frame rates for a similar level of accuracy, suggesting that our use of learned structural knowledge is a definite advantage. Running at such high speeds is beneficial since it means more measurements can be made for the same computational load, which tends to increase accuracy [18], or frees computation time for global map correction methods [17].

VI. CONCLUSIONS

We have shown that by exploiting general prior knowledge about the real world, we can derive strong structural priors, which are useful for fast initialisation of map features. This was achieved by modifying a plane-based visual odometry system to use a single-image plane detection algorithm: by detecting planes directly from a single frame, they can be
inserted directly into the map, to quickly give a concise and meaningful representation of the 3D structure. The maps we built show it is possible to rapidly extract good planar maps of scenes — something we hope to develop further, toward producing fast and accurate plane-based 3D models.

Our preliminary results also show that, by using a good initial estimate of the likely locations of planes, we achieve faster convergence and a higher frame rate. Moreover, by virtue of a more intelligent initialisation of features, and the freedom to make more measurements per frame, our method has shown the potential to reduce scale drift — this is important, since drift is a problem for all visual odometry systems, and it would be interesting to quantify the reduction in error that use of learned priors can bring. A worthwhile direction of future work, therefore, will be to see if as well producing more concise and coherent maps, using learned structural priors can give a significant and reliable improvement to the metric structure of the maps.

REFERENCES


