Independently Moving Object Trajectories from Sequential Hierarchical Ransac

Mikael Persson[a] and Per-Erik Forssén[b]
Department of Electrical Engineering, Linköping University, Sweden

Keywords: Robot Navigation, Moving Object Trajectory Estimation, Visual Odometry, SLAM.

Abstract: Safe robot navigation in a dynamic environment, requires the trajectories of each independently moving object (IMO). We present the novel and effective system Sequential Hierarchical Ransac Estimation (Shire) designed for this purpose. The system uses a stereo camera stream to find the objects and trajectories in real time. Shire detects moving objects using geometric consistency and finds their trajectories using bundle adjustment. Relying on geometric consistency allows the system to handle objects regardless of semantic class, unlike approaches based on semantic segmentation. Most Visual Odometry (VO) systems are inherently limited to single motion by the choice of tracker. This limitation allows for efficient and robust ego-motion estimation in real time, but preclude tracking the multiple motions sought. Shire instead uses a generic tracker and achieves accurate VO and IMO estimates using track analysis. This removes the restriction to a single motion while retaining the real-time performance required for live navigation. We evaluate the system by bounding box intersection over union and ID persistence on a public dataset, collected from an autonomous test vehicle driving in real traffic. We also show the velocities of estimated IMOs. We investigate variations of the system that provide trade offs between accuracy, performance and limitations.

1 INTRODUCTION

Navigation in an environment with uncoordinated, Independently Moving Objects (IMOs) is a challenging problem. A key component to the biological solution of this problem is *stereo vision*. Humans robustly identify moving objects. We do this for objects in categories never before seen, and use what is seen over time to predict where objects are moving. It should therefore be possible to find, estimate and predict moving objects in a manner useful for navigation using a stereo camera.

In this work we focus on the real-time detection and trajectory estimation of moving objects in stereo video. In the image space this gives us instance segmentation for rigid independently moving objects as shown in Figure 1. We seek the trajectories in 6D pose-space over time in order to be able to plan around them. While a robot would benefit from a multitude of sensors such as lidar and IMU, we limit ourselves to a stereo sensor. This reduces the complexity of the system and the number of dependencies.

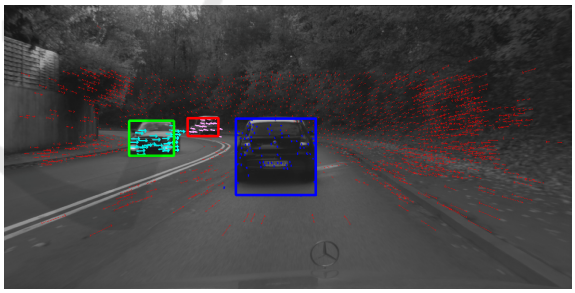

Figure 1: Shire result. World tracks in red, IMOs marked by bounding boxes along with the tracks that are considered as belonging to the objects. Their full 6D trajectories in pose space are estimated in each frame. Shire does not use deep networks or semantic segmentation.

We further limit ourselves to rigid objects, such as vehicles, as they are of key importance to autonomous driving. The principle however generalizes to piecewise rigid objects and temporarily rigid objects. This covers many real obstacles, as can be seen in Figure 4.

We present a novel, fast, accurate, and robust solution for real stereo sequences. The solution is based on a causal greedy probabilistic approach to the assignment and cluster count problems. The solution exploits the sequential nature of the track data and the

[a] https://orcid.org/0000-0002-5931-9396
[b] https://orcid.org/0000-0002-5698-5983

Persson, M. and Forssén, P.
Independently Moving Object Trajectories from Sequential Hierarchical Ransac.
DOI: 10.5220/0010253407220731
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 5: VISAPP*, pages 722-731
ISBN: 978-989-758-488-6

typical distribution of moving objects in real scenes. The proposed system extends the feature tracking type of visual odometry system (Klein and Murray, 2007; Badino et al., 2013; Mur-Artal and Tardós, 2017; Persson et al., 2015; Cvišić et al., 2017) to detect and estimate multiple trajectories. The implementation is based on CV4X (Persson et al., 2015) which achieved state of the art results on the KITTI egomotion benchmark and can thus provide reliable egomotion. We now extend this system to also estimate trajectories of independently moving objects. The proposed system uses geometric consistency over time to detect objects. This takes the form of reprojection error minimization to assign tracks and estimate object states over time. As the world is also detected as a rigidly moving object, a useful by-product of the system is the ego-motion trajectory. We call the system described in Section 3 Sequential Hierarchical Ransac Estimation (Shire). Shire performs well in practice, in particular for nearby and fast moving objects. These objects are exactly those that are important for dynamic obstacle avoidance. Shire is real-time (30 Hz) on a standard desktop CPU.

The system is evaluated using a novel dataset, which we make available (Persson, 2020). Ideally we would have liked to evaluate our method by comparing the trajectories of the estimated IMOs to their ground truth. To the best of our knowledge no suitable dataset is available for this, nor can we generate such ground truth for our dataset. The dataset is collected from our experimental vehicle, when driving in real traffic. It covers inner city, country road, and highway. We evaluate using bounding box instance segmentation of moving objects. This acts as a proxy for detection, ID persistence, and estimation. This dataset has been preprocessed and we provide rectified images, estimated disparity and semantic segmentation as well as manually annotated IMOs. At the dataset link, you can also find evaluation software, the Shire code, and an example video.

2 RELATED WORK

IMO detection and trajectory estimation could be approached by *scene-flow* methods. Scene-flow is the observed 3D motion per pixel. These methods implicitly segment the flow into rigid objects. Sceneflow approaches can be categorized as classic or deep learning based methods.

Classic methods are well represented by Piecewise Rigid Scene Flow (Vogel and Roth, 2015). The method uses classic flow, classic stereo and (classic) superpixel-segmentation. These are used to form a regularized sceneflow cost function. This cost is then optimized over a single stereo image pair using Gauss-Newton (GN). The recent Deep Rigid Sceneflow (Ma et al., 2019) method is similar in many ways. The method uses a flow network, a stereo network, and a semantic-segmentation network. The networks are used to form a similar cost function. The cost is then optimized over a single stereo image pair using GN, unrolled for differentiability. Replacing each component with its deep learning based variants requires supervised training. This implies the need for large datasets with both semantic and 3D correspondences ground truth. The cost of such is the main disadvantage of the modern method. However, comparison on the KITTI sceneflow benchmark shows that the latter method significantly outperforms the former. However, while the deep learning approach is faster at 746ms compared to 3 minutes on 0.5Mpixel images, both methods are far from real time. Both methods could hypothetically be extended to use multiple images. However, it is unclear how to do so without further increasing the computational cost. The methods are potentially useful as input to the proposed system. We conclude that they are currently to slow for our target application.

Another deep learning approach which targets a similar problem is MOTSFusion (Luiten et al., 2019). This method aims to identify and separate cars, both still and moving. The method uses deep optical flow, bounding box detections, semantic segmentation, deep stereo. The method also performs egomotion compensation using ORBSLAM. Next they use they use a per track geometrically aware bootstrap tracking method method in order to associate tracks over time. This results in strong id propagation performance for the object detections. The method also achieves good results on the MOTS benchmark. The purpose here is tracking rather than 6D trajectory estimation however, though the method could be extended or used as input. Similar to the deep rigid sceneflow the issues are with general moving objects and computational cost. MOTSFusion operating at 0.5MPixel takes 440ms per frame, and while this may be applicable in some cases, it is a long time in collision avoidance.

The good performance of the deep learning approaches comes at a price. Relying on bounding box and/or segmentation limits the systems to the classes for which data is available. For new cameras or scene content, the deep learning methods require finetuning, which requires ground truth. Even if only to adapt to the minor differences in resolution, scene and image characteristics, this requires ground truth data we do not have. By contrast, the classic methods typically

only require manual tuning of a few parameters.

A different approach to the problem of IMO state estimation is found in the 6DVision system (Rabe, 2012). The system fuses stereo and KLT tracks over time using a set of Extended Kalman filters over 3D position (with a constant velocity state). The each filter set is initialized using nearby features states. The filters of each filterset have varying state priors, measurement noise and process noise. The main advantages of this method are low computational cost. The main downside is that individual tracks are noisy and that the filter sets take a long time to converge. In practice, the state estimates are strongly smoothed. This leads to decent results for long tracks, but the system often fails to account for rapid changes.

In this work we introduce a new driving dataset for the purposes of evaluating the system. Compared to KITTI our dataset has higher resolution (2Mpixel vs 0.5Mpixel) and higher framerate (30fps vs 10fps). Using our own dataset allows us to ensure highly accurate calibration which is required for our method. We cannot evaluate on the sceneflow benchmarks such as KITTI as it requires dense sceneflow, which our method does not provide. Nor can we meaningfully submit to the KITTI (Geiger et al., 2012) MOTS benchmark. This is because the MOTS benchmark does not distinguish moving objects from stationary ones. In practice the latter overwhelmingly outnumber the former. We have however run our system on the these datasets. See section 4.3 for results and further discussion.

Real time stereo VO systems lie on a scale from direct to feature based. Direct systems estimate geometry directly from pixel intensities and pixelwise depths. Feature based systems use higher level image features such as patches and descriptors. The association process of direct systems relies heavily on assumed geometry. This is because pixel intensities are ambiguous, and the geometry constrains the problem. Relying on predicted geometry in this way is a form of geometrically supported tracking. The simplest case of which would be to search for correspondences along a predicted epipolar line. The DSO systems of (Engel et al., 2018) and the stereo variant by (Wang et al., 2017) are typical direct VO systems. A major advantage of single geometry supported tracking is that it allows the use of line features. This improves the VO systems robustness to inarticulate environments and simplifies the tracking. The downside is that it fundamentally limits this type of system. While DSO could hypothetically be recursively applied in a similar manner to our system, in practice we have found that initializing the DSO requires a single dominant motion in view which precludes this.

Shire is an extension and generalization of the well studied point feature type visual odometry system class. The basis for our method is found in systems such as PTAM (Klein and Murray, 2007), MFI (Badino et al., 2013), ORBSLAM (Mur-Artal and Tardós, 2017), Cv4x (Persson et al., 2015) and SOFT(Cvišić et al., 2017). These methods emphasize the use of inlier/outlier identification and windowed bundle adjustment. They also use measurement virtualization, keyframing, and as well as non-causal processing. Finally most point feature based systems use geometrically supported tracking. Rather than as a core of the tracker, the focus is often on improved robustness against structural aliasing. But it is also used to extend tracks which in turn reduces trajectory drift. A core assumption of these methods is that they are limited to a single rigid object in view. In turn this means that the tracking can and should detectably fail for everything else. Due to the features used, feature based systems are inherently robust against the basic aperture problem. This means that the integration with the tracker is less restrictive. For all VO systems, geometrically supported tracking is key to good performance. How limiting this is varies however. Out of the systems considered, the bootstrap tracking by matching of Cv4x, is perhaps the least restrictive. Cv4x essentially iterates selecting the best correspondence and finding the pose. This places places no restrictions on the initial search. Without the hard reliance on the single motion assumption, it is suited for our purpose. We strip down these systems into a simple core suited for fast causal local mapping and generalize this core system to support IMOs. We also investigate how to apply geometrically supported tracking to Shire with Shire-retrack.

3 METHOD

3.1 Visual Odometry Overview

Our system is an extension of the feature tracking type of visual odometry system (Klein and Murray, 2007; Badino et al., 2013; Mur-Artal and Tardós, 2017; Persson et al., 2015; Cvišić et al., 2017). Tracks are key to such systems and can be defined as the sequence of image coordinates of a repeatably localizable feature (landmark). An example would be the projections of a 3D point with distinct appearance. Landmark tracking visual odometry systems exploit that the tracks lie on a single static rigid object (world). This facilitates identifying outliers and reducing tracking errors.

First landmarks are detected and then tracked in latter frames. A cost corresponding to Eqn 1 is then minimized. This computes the egomotion trajectory $P_t \forall t$ and local 3d point cloud $x_i \forall i$.

$$\min_{P_t, x_i} \sum_{t,i} \rho(y_{t,i} - \wp_\theta(P_t x_i)) \quad (1)$$

where ρ is a robust error function, $y_{t,i}$ is the observation of landmarks x_i, at time t and P_t is the camera pose, i.e. the rigid transform, at time t for $t \in \{t_0, t_1, ..., t_N\}$ and $\wp_\theta$ is the projection operator suited for x_i where θ is the intrinsic parameters of the camera.

In the simplest case, $x \in \mathbb{R}^3$, is a 3D point in world coordinates. The observation noise is Gaussian plus outliers, the cameras are calibrated, and the correspondences are given.

Typically, ρ, is the L_2 loss combined with an outlier rejection scheme based on reprojection error. The orientation part of the pose is represented as a unit quaternion. Primarily to avoid the gimbal lock problem of Euler angles.

The cost Eqn 1 is non-convex, but optimization will generally converge if the data is balanced and well initialized. A good way to initialize is by sequentially growing the cost i.e. adding one image at a time. This is further improved by computing PnP to find the initial pose of each new frame before bundle adjustment.

3.2 Stereo Point Feature Visual Odometry

Shire is best understood as an extension to point feature visual odometry, and for that reason we will first introduce a base method.

Given a sequence of rectified, calibrated stereo images, iterating Algorithm 1 provides a basic, but good, ego-motion trajectory estimate as indicated by (Persson et al., 2015; Cvišić et al., 2017; Mur-Artal and Tardós, 2017).

Specifically we use: the SGM from (Hirschmüller, 2008) to compute disparity. The KLT implementation from (Rabe, 2012). The P3P and PnP methods from (Persson and Nordberg, 2018), and ANMS from (Gauglitz et al., 2011).

The stereo reprojection error, ε, Eqn 2 for each track includes disparity $d(y) \geq 0$. It also requires the points to be in front of each observing camera.

$$\varepsilon(y, d, P, x) = (y - \wp_{\Theta_l}(Px))^2 + \lambda(d - [\wp_{\Theta_l}(Px) - \wp_{\Theta_r}(P_{rl}Px)]_0)^2 \quad (2)$$

where Θ_l, Θ_r are the camera intrinsics, and P_{rl} is the relative pose of the stereo rig. Subscript 0 denotes using the horizontal coordinate only.

Algorithm 1: Stereo Point Feature Visual Odometry.

1: Let: Left, right images at time t: LR_t
2: Let: Tracks of object o at time t: $tracks_{o,t}$
3: Let: Pose at time t: $Pose_t$
4: # Compute disparity
5: $depth_t = \texttt{SGM}(LR_t)$
6: # Track features using KLT
7: $tracks_t \leftarrow \texttt{tracker}(LR_t, tracks_{t-1}), depth_t(LR_t)$
8: # Compute pose: Ransac P3P and refinement
9: $P_t \leftarrow \texttt{robust\_PnP}(tracks_t)$
10: $\texttt{windowed\_bundle\_adjustment}(P_:, x_:)$
11: $tracks_t \leftarrow inliers(tracks_t : \varepsilon(track_t) < \tau_{inlier})$
12: # Detect new features
13: $tracks_t \cup \leftarrow \texttt{detector}(LR_t, depth_t)$
14: $tracks_t \leftarrow \texttt{ANMS}(tracks_t)$

The parameter $\lambda = 0.5$ is a balancing weight correcting for the relative variance of the tracker and disparity methods. We found this value by comparing the relative errors when viewing a static scene. The method is fairly robust to different values however. $\lambda \in [0.1, 2]$ provides similar results if the other thresholds are adjusted accordingly.

We minimize the squared L2 norm of the stereo reprojection error, as it achieved good results and performance:

$$\min_{P_t, x_i} \sum \varepsilon(y_{t,i}, d_{t,i}, P_t, x_i)) \forall t, i \quad (3)$$

We have found that the quality and robustness of Algorithm 1 method is bounded by the tracker and disparity estimates (assuming pixel accurate calibration and typical scenes).

Algorithm 1 is not limited by geometrically supported tracking. Therefore we can extend it to support multiple independently moving objects.

Many different tracking, disparity or optical flow methods could be used, but this is not the focus of this work. Instead we use real time implementations of standard methods. It is common to use keyframing to facilitate loop closure or reduce computational cost, but for clarity this is omitted here.

Adaptive Non-Maxima Suppression (ANMS) is often overlooked in VO descriptions (Gauglitz et al., 2011). We explicitly include ANMS as it is key to good performance both computationally and in terms of accuracy. ANMS conditions the optimization problem, as nearby observations will have correlated errors. The importance of ANMS also indicates that there is a limit to the accuracy gain possible by increasing the track density. In other words, observations must cover a minimum amount of image for accurate pose estimation, regardless of track density. This applies to both Shire and Algorithm 1. Typical ANMS parameters are radius = 1% image width, 100

minimum tracks, and a strength score which is (track age, cornerness score).

We use the high performance sparsity aware Ceres Solver (Agarwal et al., 2020) to optimize the cost.

3.3 Extension to IMOs

We view reprojection error based inlier/outlier classification as a world/not world assignment operator. Thus we extend this notion to the simultaneous assignment and optimization of multiple rigid objects. A simple idea, which never the less results in a challenging optimization problem:

$$\min_{P_{c,t},x_i,c_{is}} \sum_{t,i\in c_{is}} \rho(y_{t,i},d,P_{c,t},x_{c,i}) + C(c_{is}) \qquad (4)$$

Here ρ is a robust and weighted stereo reprojection error, similar to Eqn 2. The pose $P_{c,t}$ is that of the camera in IMO c:s coordinate system at time t and $x_{c,i}$ is the 3D point coordinates in IMO c:s coordinates. The set of features belonging to object c is c_{is}. Finally the clustering penalty $C(c_{is})$ accounts knowledge of the distribution of objects. Without which a global minimum is trivially found by putting every point in its own object.

From an abstract perspective the problem is one of association, clustering and minimization. The clustering space is non-metric due to the use of reprojection errors as distance to cluster. This violates the assumptions of many greedy solver such as KMeans. Further complicating things, the number of clusters and their distribution is unknown. In short the problem is challenging.

Restricting ourselves to rigid objects, there is no fundamental difference between the estimation of the IMO trajectory with respect to the static world, and the estimation of the IMO trajectory with respect to the moving rigid object. Thus if the cluster association of each track is known, the stereo VO method of subsection 3.2 can be applied to each IMO independently.

Similar to the static world inlier/outlier case we can assign tracks to IMOs using a noise model, and when there is sufficient evidence, initialize new IMOs. The key observation on which the system is built, is that while tracker errors leave many track assignments ambiguous, an often sufficient subset can be unambiguously assigned in each frame. Thus by accumulating information over time we causally detect, assign and estimate in real time. In essence we have three possible states for a track, inlier, ambiguous, and outlier.

3.4 Shire

Given a sequence of rectified, calibrated stereo images, Shire iterates the steps in Algorithm 2. This algorithm segments the scene into rigid IMOs and estimates their full 6D pose trajectories. The variable *objects* consists of both the object pose trajectory and the 3D coordinates of landmarks associated with the object.

Algorithm 2: Shire.

Let: Objects at time t: $objects_t$
Let: Tracks of object o at time t: $tracks_{o,t}$
Let: Reprojection error of track tr for object given optimized track state: $\varepsilon_{o,tr}$
$score(o,tr) = max_t(\varepsilon_{o,t}(tr))$
$inlier(o,tr) = score(o,tr) < \tau_{inlier}$
$found(o) = \#tracks_{o,t} > IMO_{minimum\ tracks}$
Track, depth, detect (as per 3.2)
Unassigned: $us_t \leftarrow us_{t-1} \cup$ new tracks
for each $o \in objects_{t-1}$ **do**
    $poses_{o,t} \leftarrow \texttt{PnP}(objects_{0:t-1}, tracks_{o,t})$
    $tracks_{o,t} \leftarrow tracks_{o,t-1}\ in\ inlier(o,tracks_{o,t-1})$
    $us_t \leftarrow tracks_{o,t-1}\ not\ in\ inlier(o,tracks_{o,t-1})$
    $objects_t \cup \leftarrow found(o)$
    `windowed_bundle_adjustment`(o)
    $os = not\ inlier(o,tracks_{o,t})$
    if $o \neq world$ **then**
        Split Tracks: $us_t \cup \leftarrow lasttwo(os)$
    end if
end for
Let: $cs \leftarrow$ Potential IMO candidates
for each $tr \in us_t$ **do**
    $rs \leftarrow sort(score(o,tr) \forall o \in objects_t)$
    if $rs[0] \geq \tau_{inlier}$ **then** $cs_t \cup \leftarrow tr$
    else if $rs[1] > \tau_{outlier}$ **then** $tracks_{o,t} \cup \leftarrow tr$
    end if
end for

Initialize New Object(cs_t)
$cs_t.x \leftarrow disparity_{t-1}(cs_t)$
candidate: $c \leftarrow PNP(cs_t)$
inliers: $ins \leftarrow cs_t s.t. rpe_{c,cs_t}$
if $\#inliers > \tau_{creation\ count}$ **then**
    Optimize candidate trajectory and points
    $objects_{o,t} \cup \leftarrow candidate$
end if

Additional details in the source code.

3.4.1 Assignment

The track score for a trajectory is the maximum of the reprojection errors among the observations given

the optimal state. The optimal state is the state which minimizes the squared sum of reprojection errors for that trajectory over the optimization window. Computing track scores for all trajectories is comparatively costly. Thus this is only performed for tracks which are unassigned. The inlier test requires that the reprojection error satisfies the inlier threshold, and is not ambiguous with respect to the second best possible assignment. Additionally for IMOs, it is required that the point is within three meters of the object center. For an IMO we require a minimum track count of $\tau_{creation\ count} = 20$. For the inlier, and outlier thresholds we set $\tau_{inlier} = 2.5$ pixels, and $\tau_{outlier} = 4$ pixels. The thresholds were found by looking at error rates of tracks with known assignment. They coarsely correspond to a probability of $P(\text{observation}|\text{from object}) = 0.8$ and $P(\text{observation}|\text{from object}) = 0.05$.

3.4.2 Detection

The Ransac PNP solver is likely to find IMOs in size order, beginning with the first object which is considered the world. Visualization aside it is generally treated the same as the IMOs with two exceptions: Outliers which successfully pass the inlier check for the world, are assigned to it regardless of ambiguity and regardless of distance to its mass center. This prevents the formation of competing static objects should an IMO come to a full stop.

Assignment to IMOs is only considered if the IMO median world coordinate mass center is within five meters of the points world coordinates in the latest frame.

Initialization of a new object is performed as for the first two frames in stereo VO, its origin set to identity at the timestamp of the previous frame. After initialization, any object for which the tracks do not satisfy the assignment test to itself are unassigned. Objects with less than ten remaining tracks are discarded, and its tracks are split and unassigned.

IMOs typically cover a small part of the image limiting the number of tracks available. This in turn necessitates denser tracking to meet minimum robust track count for detection and estimation. To account for this we use a small ANMS radius 12 pixels Experimentally we have found that this is a good tradeoff between speed and robustness. Increasing the radius increases computational speed, but impacts the detection of smaller objects if increased above 20 pixels. Lowering the radius increases computational cost without an observed associated gain in performance.

The system initializes at most one new object each frame in order to limit max per frame processing time. Should performance permit, it is straightforward to initialize multiple objects by iterating this step.

Ransac PNP with nl-refinement is used because it is robust against the high outlier ratios that occurs when searching among candidates formed from a several not yet identified IMOs.

3.5 Computational Speed

The tracking and disparity can be performed in advance in parallel on CUDA, taking taking 30ms for 2Mpixel images.

In order to speed up optimization, bundle adjustment of trajectories is performed over a maximum of 500 tracks. These tracks are found by increasing radius ANMS. The remainder of the tracks are subsequently optimized using fixed poses. To further improve performance, all optimizations and error computations use a window size of five.

The baseline version of Shire operates at 30fps with two frames of latency between input and trajectory estimates. Shire retrack requires an additional 12ms.

3.6 Semantic Segmentation

While we generally argue in favor of the generic geometric approach, in the specific case of vehicles, fast semantic segmentation networks are available. Therefore we investigate how the method can be improved by semantic labels. In the variant of our method called Shire-labels we use the FCN (Cordts, 2017; Long et al., 2015). The segmentation network was trained on Cityscapes (Cordts et al., 2016).

In Shire-labels, only tracks that lie on vehicles can be assigned to IMOs. The semantic segmentation was performed offline in batches, but may be feasible to run in real time online. The evaluation shows a minor improvement to intersection over union (IoU) scores. Qualitatively we see a significant reduction in false positives.

4 EXPERIMENTS

We investigate four variations of the Shire system. Shire is the baseline and the fastest variant. Shire+labels uses semantic segmentation to limit the search for candidate tracks. Shire+labels only considers tracks where the starting pixel is labeled as a car or truck as IMO candidates. Shire+retrack uses a second tracking pass to refine the track associations for lost tracks. The second tracking pass uses predictions based on the estimated world and object motions. The

new track result is kept if consistent with its prediction. This retrack tracking step comes at the cost of an additional tracking pass. However, unlike the base tracking pass, this only uses the bottom pyramid level and is slightly faster. Shire+retrack+labels uses both labels and the retrack step.

4.1 Dataset and Evaluation

The lack of benchmarks for the identification of moving objects makes evaluation of IMO estimation a challenge. Object motion 6D trajectories are generally difficult to acquire for real sequences. The available benchmarks are focused on bounding box segmentation of semantic classes. This makes them ill suited for evaluating the quality of a system which needs motion, but not semantic class. We propose to evaluate the system by bounding box IoU and number of ID switches. Specifically we are interested in the results on the 2Mpixel stereo dataset we collected from a Daimler test vehicle. We annotate 7k images for quantitative evaluation.

We performed manual bounding box instance segmentation for IMOs. The criteria for an object to be annotated as an IMO were as follows: The object is a car or truck. The object boundary must move more than two pixels from frame to frame. This is after manual egomotion compensation i.e. compared to known static background. The object must be bigger than 5% of the image and closer than 70 meters. The distance was determined using manually verified disparity. Further, the object must be in unoccluded view for at least five sequential frames. Objects that come to a complete stop for more than five frames are no longer considered IMOs. Unoccluded view requires that 70% or more of the object must be visible. Any IMO which fully leaves view is considered as having a new IMO ID. Not all IMOs are annotated.

This IMO selection is conservative in the sense that the system can detect smaller, shorter lived and occluded IMOs. The IMO requirements are made in part because the annotation of such IMOs is often ambiguous. Their prevalence in the dataset would also make annotation too costly. This means that false positives cannot be automatically evaluated. As a result the primary scores of interest are intersection over union and false negatives.

Bounding boxes for each proposed object are computed in each frame as the range of the associated tracks positions. The proposals are then associated to the ground truth boxes by IoU order in a one to one greedy fashion. The greedy assignment is in descending GT bounding box area order with the added constraint of a minimum IoU score of > 0.2. The result of which is summarized in Figure 2.

Table 1: Per frame computational cost. Note that this is only the cost of the Shire algorithm. The first tracking, disparity and semantic segmentation are excluded as these can be performed in parallel one frame in advance. However, the second tracking pass required by the retrack method is included in the timing as this cannot be performed in parallel.

Method	median time	mean time
shire	27 ms	30 ms
+labels	26 ms	29 ms
+retrack	43 ms	45 ms
+retrack+labels	40 ms	42 ms

Table 2: Number of ID switches, and number of false negatives. Total nr of annotation bounding boxes is 9380. Total images 7000.

Method	switches	missed
shire	4	1430
+labels	4	1730
+retrack	3	1505
+retrack+labels	11	902

We compute ID switches as the number of times the IMO ID of the IoU order top choice for a GT IMO changes. An object must be found at least once to be included in the computation. If an object is found lost and found repeatedly again any number of frames later, each repetition counts as one ID change. The evaluation software implementation is available.

4.2 Results

Table 1 shows that our method performs in real time on the i7 3GHz CPU. Table 2 shows that the number of ID switches is infrequent, but also indicates that more id switches happen for harder to detect IMOs.

Figure 2 shows excellent performance for the combined variant. It also shows that the labeled variant is better at finding object boundraries. Without label information slightly more IMOs are found at low IoU scores. Most of these are false positives that happen to overlap with a gt object. Overall performance is fairly good for detected IMOs. Unfortunately a significant fraction are not detected. IMO shadows play a role in reducing IoU accuracy as shown in Figure 5.

The experiments show that the purely geometric Shire achieves moderate success in the generic setting. This result is significantly improved when combined with the segmentation labels, though this is hidden by partial overlaps with false positives at lower IoU scores. This result is further improved by retracking, though less so in the purely geometric case.

The main differences between the using segmentation and not lies in the effect of shadows and outliers

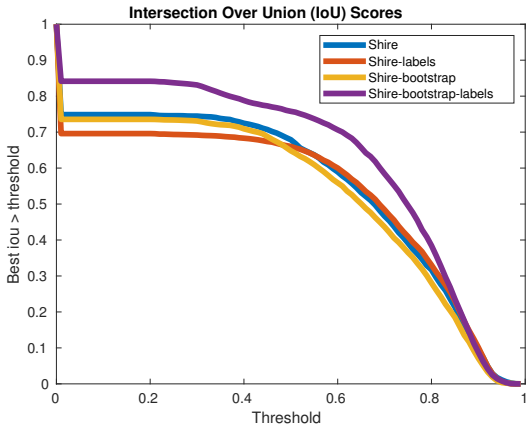

Figure 2: Intersection over Union (IoU) for the best one to one match for each GT IMO. We see that the using the label information (Red) provides good results when a high overlap is required but misses slightly more objects 30%. Bootstrap (Blue) provides a direct improvement over Shire (Yellow). Combined they achieve a significant improvement (Purple). Note that not all IMO are detected in every frame. Roughly 16-30% are not found, most of which lie at far distances at the beginning or end of a GT IMO.

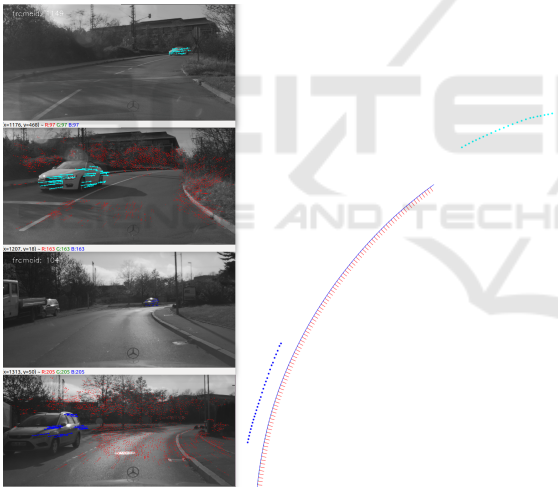

Figure 3: Shire results. On the left top to bottom. First frame where cyan car is found. Last frame before it is lost. First frame where the blue car is found. Last frame before it is lost. On the right. The two cars mass centers in world coordinates over time shown as dots and the Vehicle egomotion poses shown as blue/red coordinate systems. Egomotion is from the bottom left to the upper right, IMOs are driving in the opposite direction. Best viewed electronically.

on bounding boxes. There is also a small impact on speed and an increase in the number of false positives.

A simple heuristic was used to reduce the impact of outliers that fit the geometry. Tracks used for finding the bounding box must be tracked for at least three frames. This heuristic applies only if the at least half

Figure 4: Shire result. A pedestrian IMO detected by the system. While violating the rigid object assumption, people, and most other non rigid IMOs, are often locally rigid. At 100x200 pixels the pedestrian is too small to be fractured into several IMOs. Instead it is intermittently lost and re-detected. The ability to detect generic IMOs is an advantage of the purely geometric method.

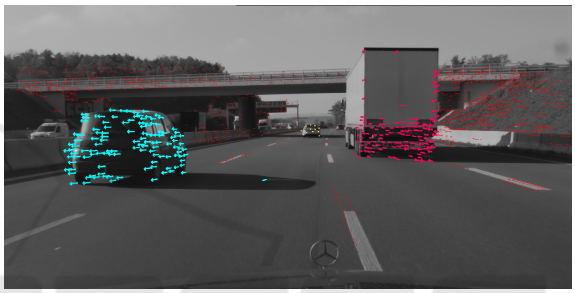

Figure 5: Shire results. Red tracks are world, cyan, yellow and pink are IMOs. The track on the cyan car's shadow causes the IMO bounding box to be significantly extended for the car. This happens because the sun is distant, and the car moves in the plane the shadow is projected on top. A downside of the purely geometric method, as we do not collide with shadows.

the tracks on the object have been tracked for three frames. We note that better heuristics could be devised. In particular an analysis of the plane of the motion could be used to identify tracks on shadows. However this is left as future work.

We observe that the system excels at detecting and segmenting fast or nearby objects, but struggles with distant ones. This expected result is never the less beneficial in terms of which objects are important for safe navigation.

Like most feature tracking VO systems, the method is bounded in performance by the tracker. The tracker struggles if the object tracks lengths near or exceed the maximum supported distance. This is in part due to direct search distance limitation and due to large appearance changes due to perspective. The feature detector also struggles with low contrast or blurry images, such as caused by dark regions in otherwise sunny scenes and rain.

The system provides decent ID propagation for IMOs. The primary error source being when two

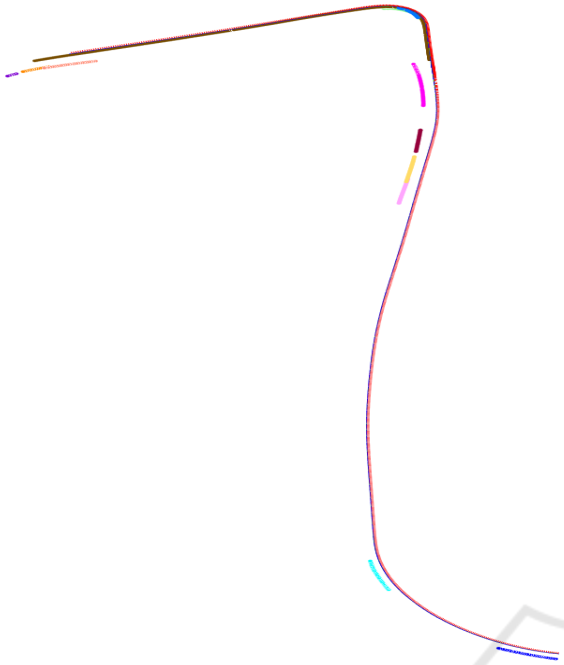

Figure 6: Starting bottom left, the RGB coordinate axis shows the car trajectory. The colored dots show the various IMOs encountered and their estimated trajectories. Some are driving in the lane to the left, others along the same lane. Best viewed electronically.

IMOs cover separate parts of and therefore compete for a single GT object. More concerningly the system fails to detect a fair number of the annotated objects. There are two main causes for false negatives. The first case is a insufficient deviation from the world geometry tracking. This happens for objects that are too far away, especially if they are driving away from the observer. A single car driving away in the same lane at a distance above $35m$ results in nearly a third of all missing detections. The second case is that the tracker or stereo has failed, which happens for objects that are moving or warping too fast in the image. This is a fundamental failure of the tracker and could be addressed by better tracking.

4.3 MOTS and KITTI Results

Shire sample results on the MOTS and KITTI sequences are shown in Figure 7. The Shire parameters are based on tracker and disparity system properties. Therefore we expect they should work for the MOTS and KITTI dataset as well as our. Thus the same parameters are used throughout.

Qualitatively the performance is slightly worse compared to results on our own dataset. Primarily in terms of the number of false positives. This may be partially explained by the frame-rate difference, as the

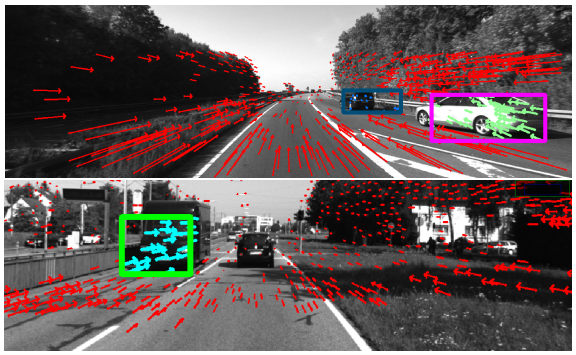

Figure 7: Top image shows performance on the KITTI benchmark, bottom image shows performance on the MOTS benchmark. Note that the color image from MOTS was converted to grey scale for illustration purposes.

expected parallax increases. Though the difference in sensor characteristics, baseline and resolution may be contributing. We also note that performance is subpar to the MOTSFusion system, though we do note that our system is more than ten times faster. The performance on the MOTS varies significantly between sequences, but not by scene complexity as may be expected. A possible explanation is that the calibration is slightly worse for some sequences. A calibration error of 1-3 pixels instead of 0-1 pixel error would be sufficient, as Shire is sensitive to calibration errors due to the assignment mechanism. This problem does not appear to apply to the KITTI sequences which use the grey scale cameras instead of the RGB cameras. Since the grey cameras were calibrated for the VO challenge, but the RGB cameras have primarily been used for deep learning, we would not be surprised if they are better calibrated.

4.4 Conclusion

We have presented stereo based Sequential Hierarchical Ransac Estimation (Shire). Shire detects and estimates, in realtime, the 6D pose trajectories of multiple moving objects, and the egomotion using geometric consistency alone. The system can benefit from semantic segmentation, but, unlike for similar methods, it is not required. Shire can be used to predict object trajectories for path planning with dynamic obstacles. It can also detect spatiotemporally separable objects, which form an interesting semantic class, or be used to generate ground truth for optical flow including that on moving objects. Though we leave the training of self supervised systems using Shire as future work. We provide the core Shire code, and the dataset used for evaluation, as well as annotation and evaluation tools (Persson, 2020).

REFERENCES

Agarwal, S., Mierle, K., and Others (2020). Ceres solver. http://ceres-solver.org.

Badino, H., Yamamoto, A., and Kanade, T. (2013). Visual odometry by multi-frame feature integration. In *First International Workshop on Computer Vision for Autonomous Driving at ICCV*.

Cordts, M. (2017). *Understanding Cityscapes: Efficient Urban Semantic Scene Understanding*. PhD thesis, Technische Universität, Darmstadt.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cvišić, I., Ćesić, J., Marković, I., and Petrović, I. (2017). Soft-slam: Computationally efficient stereo visual slam for autonomous uavs.

Engel, J., Koltun, V., and Cremers, D. (2018). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625.

Gauglitz, S., Foschini, L., Turk, M., and Hollerer, T. (2011). Efficiently selecting spatially distributed keypoints for visual tracking. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 1869–1872.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hirschmüller, H. (2008). Stereo processing by semi-global matching and mutual information. *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:328–341.

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Luiten, J., Fischer, T., and Leibe, B. (2019). Track to reconstruct and reconstruct to track.

Ma, W.-C., Wang, S., Hu, R., Xiong, Y., and Urtasun, R. (2019). Deep rigid instance scene flow.

Mur-Artal, R. and Tardós, J. D. (2017). ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.

Persson, M. (2020). IMO dataset. https://www.cvl.isy.liu.se/research/datasets/imo-dataset/.

Persson, M. and Nordberg, K. (2018). Lambda twist: An accurate fast robust perspective three point (p3p) solver. In *The European Conference on Computer Vision (ECCV)*.

Persson, M., Piccini, T., Mester, R., and Felsberg, M. (2015). Robust stereo visual odometry from monocular techniques. In *IEEE Intelligent Vehicles Symposium*.

Rabe, C. (2012). *Detection of Moving Objects by Spatio-Temporal Motion Analysis: Real-time Motion Estimation for Driver Assistance Systems*. Südwestdeutscher Verlag für Hochschulschriften. ISBN 978-3-8381-3219-8.

Vogel, C. and Roth, S. (2015). 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, 115.

Wang, R., Schwörer, M., and Cremers, D. (2017). Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras. In *International Conference on Computer Vision (ICCV17)*.