RWTH Aachen

Department of Computer Science
Technical Report

Sequence Level Salient Object Proposals for Generic Object Detection in Video

Esther Horbert, Germán Martín García, Simone Frintrop, and Bastian Leibe

ISSN 0935–3232 · Aachener Informatik-Berichte · AIB-2014-06

RWTH Aachen · Department of Computer Science · March 2014

Sequence Level Salient Object Proposals for Generic Object Detection in Video

Esther Horbert[1], Germán Martín García[2], Simone Frintrop[2], and Bastian Leibe[1]

[1] RWTH Aachen, Germany
{horbert, leibe}@vision.rwth-aachen.de
[2] University of Bonn, Germany
{martin, frintrop}@iai.uni-bonn.de

Abstract. In this paper, we propose a novel approach for generating generic object proposals for object discovery and recognition in continuous monocular video. Such proposals have recently become a popular alternative to exhaustive window-based search as basis for classification. Contrary to previous approaches, we address the proposal generation problem at the level of entire video sequences instead of at the single image level. We propose a processing pipeline that starts from individual region proposals and tracks them over time. This enables to group proposals for similar objects and to automatically filter out inconsistent regions. For generating the per-frame proposals, we introduce a novel multi-scale saliency approach that achieves a higher per-frame recall with fewer proposals than current state-of-the-art methods. Taken together, those two components result in a significant reduction of the number of object candidates compared to frame level methods, while keeping a consistently high recall.

1 Introduction

The field of visual object recognition is currently undergoing a major paradigm shift. There has been tremendous progress both on an image classification [KSH12] and on a category detection level [FGMR10, DRS+13] and approaches are now available that can reliably detect a small number of object categories in very complex scenes [FGMR10] or that can recognize the most prominent objects in web images from a large number of classes [KSH12, DRS+13]. Still, the recognition problem is far from solved. Ironically enough, this is most visible when considering the problem of recognizing everyday objects in a continuous video stream that roughly emulates what a human sees when moving through a scene (see Fig. 1). In such a scenario, there are simply so many possible objects that it is hard to come up with an exhaustive set of categories for which specific detectors could be trained. In addition, those objects are typically not the central motive of a photograph (as in many current recognition benchmarks [EGW+10, DDS+09]), but they may be just another (small) part of a cluttered scene. As a result, the hitherto dominant paradigm of window-based classification coupled with exhaustive search is reaching its limits.

A recent trend is therefore to invert the recognition pipeline and first generate a set of category independent object proposals [ADF12, BRR12, CS12, EH14, MGV13, UvGS13] to support and guide object search [FSU13, UvGS13]. Such proposals have been shown to be useful for, *e.g.*, improving detection [FSU13], self-paced learning [LG11], or unsupervised category segmentation [KGF12]. However, most of the above-mentioned approaches are relatively unspecific – in order to achieve a high recall, they often need to generate hundreds of proposal

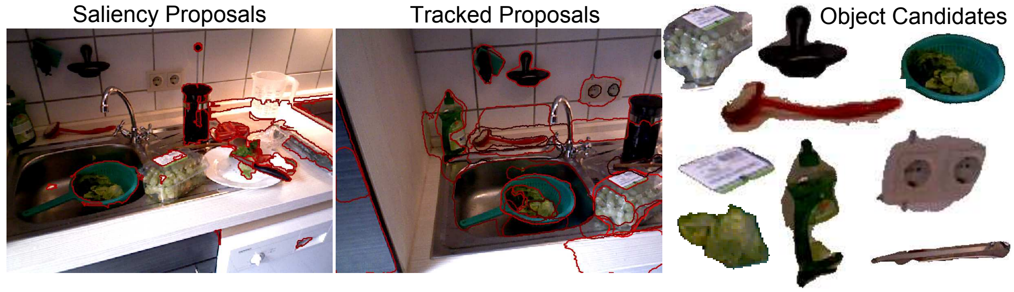

Fig. 1: Overview of our sequence-level proposal detection. Left: proposals from our frame-based, multi-saliency object discovery. Middle: sequence-level proposals from tracking. Right: Visualization of some of the candidate objects.

regions per image. In contrast, saliency-based approaches have been proposed with the goal of finding and segmenting a single, prominent object in web images [LYS+09, AS10, KF12, YXSJ13, YZL+13]. Those approaches typically generate proposal regions that adhere better to object boundaries, but their saliency formulation limits them to finding one or at best very few objects in an image; they cannot be applied for finding *all* objects in a cluttered scene.

In this paper, we want to think this trend further. As object proposal generation methods become increasingly accurate, we envision that large-scale recognition from video will be seen more and more as a retrieval problem, where a low-level module processes the incoming video stream and generates object proposals that are sent as queries to a (possibly cloud-based) recognition service. Concrete application scenarios for such a service could be a wearable camera (*e.g.*, a Google Glass like device) that recognizes objects in the user's field of view or a mobile service robot that performs everyday tasks in people's homes. In such a setting, it is not necessary that every object is recognized in every video frame. Rather, the number of recognition queries will quickly develop into the main cost factor. It is thus desired that this number be as low as possible, while covering all relevant scene objects by at least one query.

We therefore propose to address the object proposal generation problem on the level of entire video sequences. Instead of generating a large set of proposal regions for each frame, we are interested in reporting a small and consolidated number of object proposals for an entire video. For this, we take advantage of the temporal coherence of video input in order to track and evolve region proposals over time. We start from a set of region proposals for each frame and build upon a fast segmentation based low-level tracker [BR08] to propagate each of those regions independently over the next frames. Tracking fulfills a dual purpose in this procedure. First, it allows us to link independently extracted proposals from different frames that correspond to the same object. Second, it acts as a natural filter to improve proposal quality. Single-frame proposals often extend beyond object boundaries due to bad color contrast or suboptimal object viewpoint. When a tracker is initialized to such a proposal region, it will quickly diverge when the camera moves due to parallax effects. We take advantage of this effect through a series of consistency checks, terminating bad tracks already at an early stage. We then rank the remaining region tracks using shape, contrast, and tracking quality criteria and condense them into a set of sequence-level object proposals.

As our experiments will show, this procedure results in a great reduction in the number of object candidates at a consistently high recall.

In detail, this paper makes the following contributions: 1) We propose the first approach for sequence-level object proposal generation from video. Starting from a set of candidate regions extracted from each frame, our approach tracks each object candidate independently over time and subjects it to a series of consistency tests. As a result of this procedure, it can group proposals that pick out the same object and select the best representative among them. In addition, it can take advantage of camera motion to filter out region proposals that do not correspond to object boundaries. 2) In order to generate the per-frame object proposals, we present a novel method based on multi-scale saliency that achieves a higher per-frame recall with fewer proposals than current state-of-the-art methods [ADF12,MGV13]. 3) We demonstrate that the combination of those two approaches results in a concise scene summary consisting of a small number of high-quality sequence-level object proposals that could be used as queries to a recognition service. 4) We present a new benchmark dataset for object discovery from video consisting of very challenging video sequences of cluttered scenes with detailed object annotations and use this dataset to compare our approach to the state-of-the-art. We will make the dataset publicly available upon publication.

The paper is structured as follows. The next section discusses related work. Sec. 3 then gives an overview of our approach and highlights the main design goals. Sec. 4 presents our saliency based object proposal generation approach, after which Sec. 5 describes the proposed pipeline for tracking and consolidating object proposals over time. Experimental results are reported in Sec. 6.

2 Related Work

Unknown object segmentation. The capability to detect and segment unknown objects is of considerable interest for many applications in mobile robotics [STS11], autonomous vehicles [WPN12] and general visual scene analysis [KMFF13]. Many approaches in those areas either assume an active camera [BK10,KK13] or make use of 3D cues from an RGB-D sensor [BRR12,KMFF13, BBK13]. Our focus is on extracting object hypotheses from the video stream of a moving, monocular camera (*e.g.*, from a Google Glass-like setup), where we cannot control the camera motion and we do not have ready access to 3D information.

Object Proposal Generation. Object proposal generation approaches [ADF12, CS12, EH14, MGV13, UvGS13] proceed by sampling a set of candidate regions and ranking them according to their "objectness", *i.e.*, to the likelihood that the region corresponds to a full object. [ADF12] randomly sample bounding boxes to define the candidate regions and rank them using a Naive Bayes framework combining global saliency, color contrast, edge density, and location cues. [CS12] generate multiple figure/ground segmentations by solving a constrained parametric min-cuts problem from a grid of seed points and learn a ranking classifier based on Gestalt cues. [EH14] create occlusion boundary based seed regions [HEH11] and group them using a learned affinity measure between regions. They then use structured learning based on appearance features and overlap penalty terms for ranking. [UvGS13] follow a similar strategy, but use a variety of complementary grouping criteria and color spaces starting from superpixels by [FH04] to sample more diverse proposals in a selective search for object hypotheses. [MGV13] also start from superpixels [FH04] and randomly group connected superpixels by sampling partial spanning trees that have high sums of edge weights. The main effort in those approaches is spent on learning a good prediction model to rate the "objectness" of a segment based on a set of pixel or region based cues. In our work, we use the much simpler idea that the content of a good segment should differ from the content of the surrounding region. This is a property that is captured by the center-surround contrast that is at the heart of saliency approaches.

Object Saliency Criteria. Saliency and visual attention have been intensely investigated for decades in human perception [Pas97] as well as in computer vision [BI10, BTSI13]. While early computational models have been mainly designed to simulate human eye movements [IKN98], interest has recently increased to use saliency for object proposal generation as a pre-processing step for classification. However, the main focus has so far been on web images [LYS+09, AS10, KF12, YXSJ13, YZL+13], which often exhibit photographer bias. Many saliency methods have taken advantage of the special properties of such images, *e.g.*, that objects are often large and seldomly intersect with the image borders [AS10, SWLL13]. In our application, such assumptions fail and it is necessary to focus on methods that work well without them.

The key element of saliency methods is usually a measure of center-surround contrast. While this was traditionally addressed with biologically inspired Difference-of-Gaussian methods [IKN98], several other methods were recently proposed to compute this contrast. For example, [BT09, KF11, KF12] use information theory to compute the difference between center and surround distributions. Other methods compute the center-surround contrast on superpixels [PKPH12]. This is especially useful when solving the task of detecting salient objects, which is a combination of saliency computation and segmentation. Other approaches address this task by applying a segmentation method to salient blobs, *e.g.*, Graph Cuts [AS10].

A main limitation of the above approaches is that they are targeted at finding the most salient object in a scene in the sense of a global pop-out measure. In order to make saliency usable for our purpose, we therefore need to adapt the saliency formulation such that it yields proposals covering *all* objects in an image. We achieve this using a novel multi-scale saliency described in Sec. 4.

3 Overview of our Approach

The goal of our approach is to generate a concise set of object proposals that cover the relevant objects in a cluttered scene (see Fig. 1). Contrary to the existing literature [ADF12, BRR12, MGV13], however, we address the problem at the sequence instead of at the frame level. Thus, the main conceptual novelty is in how per-frame region proposals are used to obtain sequence-level results.

Fig. 1 outlines the main steps of our approach. For each frame of the input video sequence, we extract the top-k proposal regions using the multi-scale saliency method described in Sec. 4. We then track all proposals over time and use the tracker confidence as an additional proposal quality criterion. If a tracked region stays consistent over significant camera motion, this is a strong indicator that the tracked region contour indeed corresponds to a valid object boundary.

After each frame, we check for duplicates and allow new region proposals to supersede existing ones if they exhibit a better quality score. Thus, our approach builds up a set of region trajectories that span different viewpoints, increasing the chance for valid objects to be nicely delineated from their neighbors. From the final set of trajectories, we then report the best-scoring candidates and select a representative view for each of them that could be used as a query for recognition. The following sections flesh out this pipeline in more detail.

4 Generating Object Proposals

In this section, we introduce our new method to generate object proposals based on a combination of segmentation and saliency. Compared to current state-of-the-art methods, our approach achieves a higher per-frame recall with fewer proposals, which is especially important for our application of video-based proposal selection. The object proposals are generated in three steps: first, we segment the image into perceptually coherent regions (superpixels); second, we compute a saliency map that highlights salient image regions; finally, saliency is used to select and combine superpixels that form an object proposal. An overview of the approach is shown in Fig. 2.

The combination of segmentation and saliency corresponds to human perception, where first, so-called *proto-objects* [Ren00] are detected by segmentation processes that bundle parts of the visual field and that are believed to exist on all levels of the visual system [Sch01]. Second, these proto-objects are combined by focused attention (*e.g.*, saliency) to form coherent objects [Ren00].

Superpixel Segmentation. We use the popular graph-based segmentation method by Felzenszwalb and Huttenlocher [FH04] to obtain perceptually coherent segments (superpixels) that form the basis for our object proposals. We chose the parameter k, which determines the scale of observation, to slightly over-segment the image, since assembling superpixels is later accomplished by saliency selection. An example segmentation obtained with this method is shown in Fig. 2(b).

Multi-Scale Saliency Computation. For our target applications, we have to choose a saliency method which operates on video data and does not incorporate elements such as center bias or background priors based on image boundaries. Additionally, we are interested in methods that have the potential to run in real-time. Since state-of-the-art saliency methods are usually only able to detect one or a few objects per frame, we also have to extend the method in a way that enables us to detect many, preferably all, objects in a cluttered scene.

As the basis for our saliency computation, we chose the method proposed in [FMC14], which has been shown to outperform seven other state-of-the-art saliency methods. It is an adaption of CoDi-Saliency [KF12] with several improvements (new center-surround ratio, Gaussian instead of DoG pyramid, different distance measure). The method has the advantage that it computes precise saliency maps, it works for large as well as for small objects, it is applicable to web images as well as to video frames, and it is real-time capable. Briefly stated, the saliency computation works as follows. Operating on a scale-space structure (Gaussian pyramid, 2 scales and 4 octaves), computations are performed for intensity and color features. For both, center and surround contrasts are computed for different sizes at each pixel location, where color features are computed in an

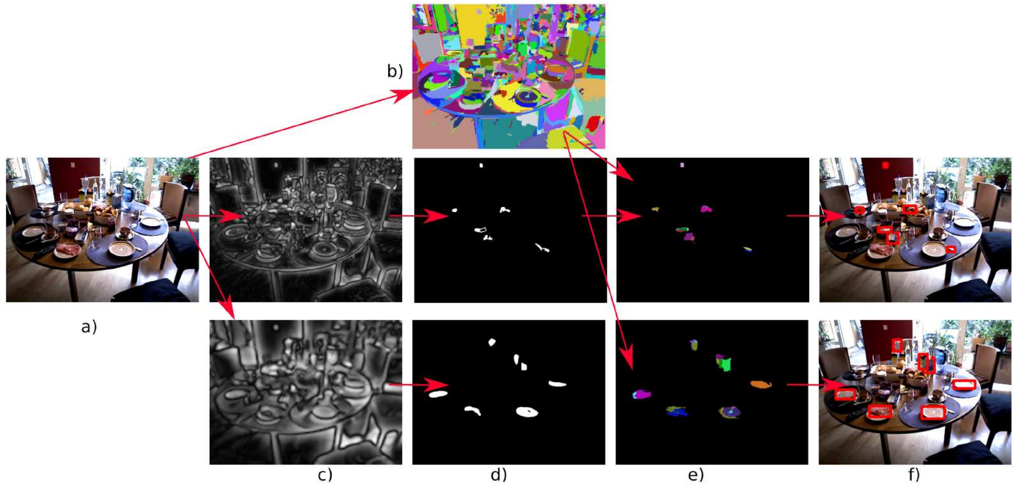

Fig. 2: Object proposal generation: a) original image, b) superpixel segmentation c) for octaves 1 and 2, their specific saliency maps, d) salient blobs obtained by region-growing, e) combining superpixels into object proposals with help of salient blobs, and e) bounding boxes of the object proposals.

opponent-color space with a red-green and a blue-yellow axis. While the original CoDi-system is based on normal distributions that represent center and surround regions and are compared with the W_2-distance, our adapted approach simply computes the Manhattan distance of the mean values of the distributions, which corresponds effectively to a Difference-of-Gaussian approach (detailed explanation in [FMC14]). As shown in [FMC14] this results in comparable quantitative results in terms of precision and recall, while the computation is faster and the saliency maps are cleaner and less blurry. This is of special importance when using saliency to extract object proposals.

In this paper, we now extend the method from [FMC14] to a multi-scale approach with split octaves. For this, we regard the octaves of the scale-space structure independently instead of fusing them into a single saliency map. Since objects of different sizes will achieve the strongest response at different octaves, this enables us to detect nested proposals. If, for example, an apple lies on a plate, ideally one scale octave will highlight the apple and another one the plate, resulting in proposals for both objects. Thus, the octave-specific saliency maps result in more proposals, finding also difficult objects which are missed otherwise. In the following, we call the octave-specific saliency maps S_l, with layers (octaves) $l \in [1, .., 4]$, where each map is the sum of the two scale maps of that layer. Examples of two octave-specific saliency maps can be seen in Fig. 2(c). In Sec. 6, we will show that while the single saliency map approach achieves a higher recall for few proposals per frame, we achieve higher recall values for a larger number of proposals. Thus, especially when aiming at finding all objects in a cluttered scene, the split-octave method is preferable.

The next step for proposal generation is to extract salient blobs from the saliency maps. In contrast to current state-of-the-art saliency methods, we are interested in finding a large number of objects per frame. Thus, simple thresholding of the saliency maps, which works reasonably well for images with one or a few prominent objects per frame, is not sufficient. Instead, we extract several blobs of different extents at each local maximum in the maps S_l. Thus, we deter-

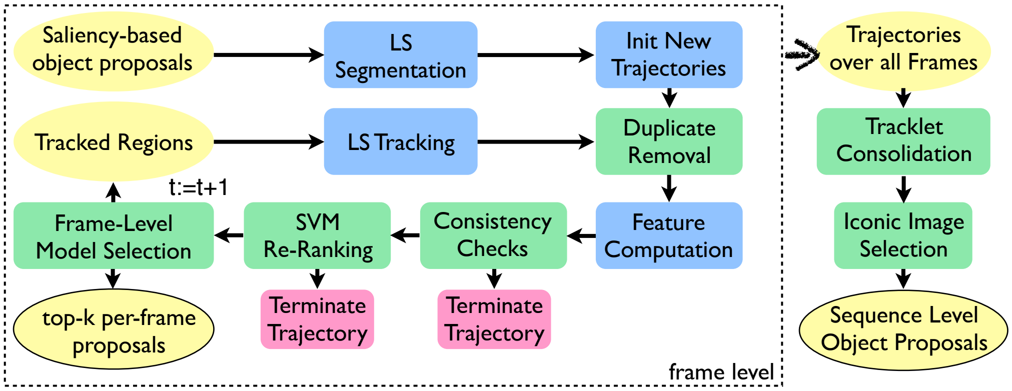

Fig. 3: Overview of the tracking pipeline for obtaining sequence-level proposals.

mine the local maxima $\{m_1, ..., m_N\}$ within each octave-specific saliency map S_l, where $m = (m_x, m_y)$. After ranking the maxima by their saliency $S_l(m_x, m_y)$, seeded region growing [AB94] is applied to each of the maxima, starting from the most salient one, to obtain salient blobs. We denote the salient blob of m as B_m. The region growing step takes each maximum m as a seed and recursively investigates its neighboring pixels: if the saliency of a neighbor pixel p is above a threshold, the pixel is assigned to B_m. The threshold is set relative to the saliency of m. That means, p is assigned to B_m if $S_l(p_x, p_y) \geq t \cdot S_l(m_x, m_y)$. We use three different thresholds $t = 0.5, 0.6$, and 0.7. The result of this method is a set of three nested salient blobs of different extents for each local maximum. This method differs from [FMC14], where adaptive thresholding is used to extract blobs from the saliency map. As our experiments will show, the region growing approach obtains more proposals, with higher precision as well as higher recall, when considering more than about 30 proposals per frame.

Combining the salient blobs of all maxima, we obtain for each octave a set of salient regions which are used in the next section to form object proposals. Some of the salient blobs are displayed in Fig. 2(d).

From Superpixels to Object Proposals. To generate object proposals, we combine several superpixels based on their overlap with the salient blobs from the previous section. Every superpixel that is covered by a salient blob by at least 30% is chosen to belong to the proposal formed by this salient blob. Next, we perform a non-maximum suppression step where we discard proposals that overlap strongly with other proposals (more than 80% in both ways). Some examples of proposals that are obtained this way are shown in Fig. 2(e).

Finally, the set of all proposals for one image is obtained by combining the proposals of all octaves. Since a saliency value can be assigned easily to each proposal by computing the average saliency of the corresponding salient blob, the proposals can be ranked. This is especially useful for applications in which real-time criteria require a prioritization of proposals.

5 Tracking object proposals

Fig. 3 shows an overview of the proposed tracking pipeline for generating sequence-level object proposals. For each frame, we initialize new tracks using the N highest scoring object proposals. Since two tracks might end up covering the same area, we check for duplicates and merge tracks overlapping by more than 70%. Next, we compute a set of quality features for each tracked region combining appearance cues and tracking confidence. We terminate inconsistent tracklets, *i.e.*, regions which have become too small, too big, have moved too fast, or have a very low tracking score. Using an RBF-kernel SVM, we then score and re-rank all remaining tracklets. Tracklets which score negatively multiple times in a row are also discontinued (this counter is reset if there is a new proposal for the same region). Whenever intermediate results are required for evaluation, we perform frame-level model selection (non-maximum suppression based on bounding box overlap) to obtain the top-k proposals. After performing this procedure for an entire sequence, we consolidate tracks by merging tracklets showing the same object and selecting an iconic representative view for each remaining track. We then report sequence-level object proposals ordered by their SVM score.

5.1 Level Set Segmentation and Tracking

For tracking reqion proposals, we use the segmentation-based level set approach described in [BR08] with the second-order optimization for the tracking component from [BR10]. This probabilistic framework segments and tracks regions using their color distribution. It has been shown to be very fast and robust to motion blur, appearance changes (*e.g.*, due to viewpoint variations) and rapid camera movement, as well as to automatic initializations [MHEL10]. It is particularly suitable for our task, since it does not only track the position, but also the region of the target object. The tracked segmentation is adapted in every frame to account for viewpoint changes and non-rigid deformations. Our re-implementation is able to track and re-segment a single region at approximately 40 fps.

LS Segmentation. Starting from an initialization region, the object is first segmented. Foreground and background probabilities P_f and P_b are modeled with color histograms and the contour is described with a level set embedding function $\mathbf{\Phi}$, which is evolved to optimize the energy functional from [BR08].

LS Tracking. The object's location is modeled as the position $\mathbf{p}$ of the *object frame*, a rectangular region around the contour, described by the parameters of a warp that transforms the object frame into the image. As in [BR08], we choose the warp to include *translation*, *scale*, and *rotation* to cope with camera motion. In each frame, the object is tracked by performing a rigid registration of the contour, such that the foreground and background model optimally match the image content. We define the tracker confidence as follows:

$$conf(T_j) = \sum_{i \in fg(T_j)} P_f(\mathbf{x}_i) + \sum_{i \in bg(T_j)} P_b(\mathbf{x}_i), \tag{1}$$

where $P_f(\mathbf{x}_i)$ is proportional to the probability of pixel $\mathbf{x}_i$ belonging to the foreground (frequency of the pixel's color in the foreground color model), P_b analogous for background. For more details please refer to [BR08].

Trajectory Initialization and Duplicate Removal. In each frame, we use N object proposals to initialize new tracklets. We filter out proposals that are thinner than 10 pixels and initialize the LS segmentation with a region 4 pixels larger than the proposal. In case there already is a tracklet which strongly overlaps with the new proposal, no new tracklet is started. Moreover, since tracklets can evolve to the same region, we merge tracklets that overlap significantly. In our implementation we use an overlap threshold of IOU $> 70\%$.

Frame level features	Track level features
Object dimensions: width, height, aspect ratio.	**Tracker confidence**: *c.f.* Section 5.1.
Color contrast: χ^2 distance of color histograms [ADF12]	**Bwd tracker confidence**: when tracking the current region backwards into the previous frame.
Region symmetry: Maximum overlap of both contour halves (over 10 rotations of center axis).	**Bwd tracking overlap**: Overlap between last contour and backwards-tracked contour.
Color symmetry: Minimal χ^2 distance of color histograms (over 10 rotations)	
Contour convexity: #pixels in region / #pixels in its convex hull.	
#Non-empty bins in color histogram.	

Table 1: Frame level and track level features used for ranking tracked regions.

5.2 Consistency Checks and Proposal Re-ranking

We maintain proposal quality in two stages. We make the assumption that correct object regions can be tracked, but not each region which can be tracked is an object. In the first stage, we simply terminate tracklets that have degenerated and are clearly not tracking a consistent region anymore. This stage makes no decision about whether the tracked region is an object or not, it simply finds failed tracks. In the second stage, we use an SVM classifier to score and re-rank each tracked region and filter out low-scoring tracklets.

Region Quality Features. Ideally, we would like to use the saliency scores from the proposal generation stage also for judging the quality of tracked regions. However, the absolute saliency scores are incomparable between frames. We therefore compute a larger set of features which are used in both stages. These can be divided into frame-level and track-level features, as shown in Tab. 1.

Consistency Checks. There are several criteria by which failed tracklets can be identified. If the tracking algorithm cannot distinguish foreground from background or has lost the target, the typical behavior is extreme scaling or movement of the contour. Thus we sort out regions which become very small, very big, or which moved very fast. We also discontinue tracklets whose tracking score is below 0.7 or whose convexity score is below 0.3. Moreover, we perform backward tracking, *i.e.*, we track the region one frame into the past to see if it ends up on the same region it was coming from. If the overlap of the region in the last frame and the backwards tracked region is too low (below 0.4) or the backward tracking score is too low (below 0.7) we also discontinue the track.

SVM Re-Ranking. Since tracks might drift or fail at some point, each frame in a track is scored independently. We train a Gaussian RBF kernel SVM to classify regions into objects and non-objects. Since the data we want to classify can only be computed from tracklets, we create the training feature vectors by running our system without any duplicate removal and with less strict consistency checks on a separate training set. We initialize new tracklets from the saliency object proposals in every 5^{th} frame and track all proposals independently until the tracklets are terminated. This results in a large number of tracklet frame characteristics with both positive and negative examples. We then train the SVM using cross-validation and a grid search for the SVM parameters.

When applying our approach to a new test sequence, we use the SVM to score each tracked region. If the classification is negative more than M times in a row, the tracklet is discontinued. Whenever there is a new proposal that is a

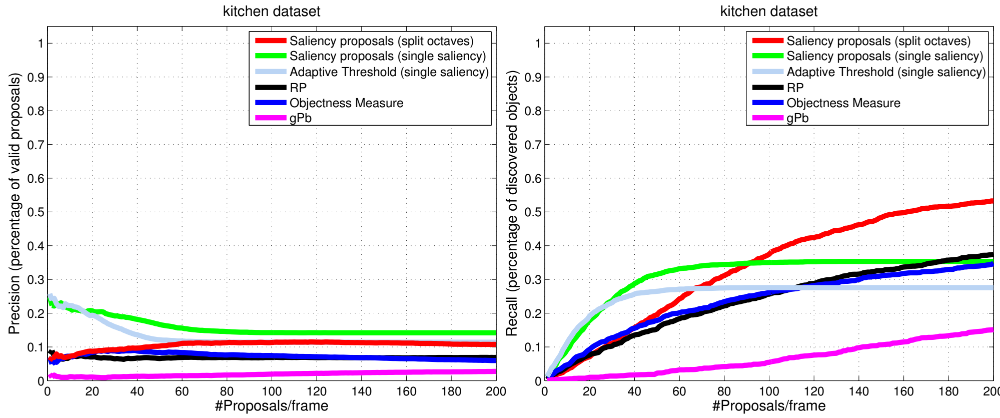

Fig. 4: Precision and recall per frame averaged over all sequences. Red: our saliency proposals, compared to two internal (green and pale blue), and three external baselines (black, blue and pink).

duplicate for a tracked region, the tracklet's counter is reset in order to avoid periodic re-initialization of tracklets for the same region.

5.3 Sequence-Level Proposal Selection

After performing tracking for the entire video sequence, we first merge tracklets that show the same object across small tracking gaps using the approach by [KPB+05]. Finally, we rank the resulting tracks by their SVM scores and report the top-ranking results. For visualization, we select a representative view of each tracked region as the frame with maximal SVM score. Altogether, this results in a massive reduction in the number of object proposals compared to the frame-level input, since each track can now be represented by a single proposal. Fig. 6 shows some sequence-level proposals that can be obtained by our approach.

6 Evaluation

Dataset. We introduce a new benchmark dataset for the evaluation of object discovery methods from video. It consists of five challenging video sequences recorded in real-world indoor environments containing a high degree of clutter. The sequences have on average about 600 frames and contain up to 80 objects. In contrast to many popular benchmarks [EGW+10, DDS+09], our dataset contains real-world images without photographer bias and with a large amount of objects and clutter. In each frame, there are on average 23 objects visible, but some views contain up to 43 objects. Object ground truth in terms of pixel-precise binary maps was annotated manually on every 30th frame, keeping the identity of objects over frames. This makes it possible to evaluate on a sequence level. We will make the dataset and the annotations publicly available upon publication.

In the following, we will first evaluate our new saliency-based object proposals, and second the sequence-level proposals obtained by tracking over frames.

Saliency Proposals. We first evaluate the quality of our saliency-based proposals with some internal and external baselines. Our proposed method is the saliency computation with region growing and split octaves (red curve in Fig. 4). As internal baselines, we used the method from [FMC14] that computes a single saliency map with adaptive thresholding (pale blue in Fig. 4), as well as an extended version of this method with region growing, but still with a single

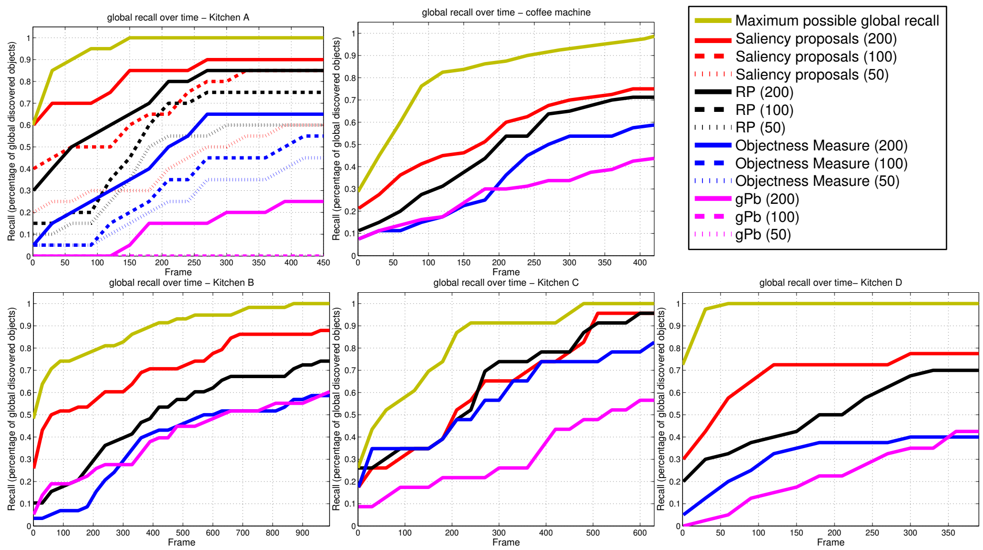

Fig. 5: Global recall over time for all sequences: from top left to bottom right, *kitchen A*, *coffee machine*, and *kitchen B,C* and *D*. Gold: maximum theoretical recall. Red: our saliency proposals. Black, blue and pink: external baselines.

saliency map (green in Fig. 4). We measure precision and recall of proposals on a per-frame level, where precision is the percentage of proposals that corresponds to a ground truth object (valid proposals), and recall is the percentage of ground truth objects that were discovered by the method. Precision and recall are plotted as a function of the number of proposals per frame. Fig. 4 shows the average results over all sequences. The results show that the new region growing method outperforms adaptive thresholding consistently in precision and recall. The region-growing, single saliency method is slightly higher than the split-octave method in terms of precision, and also recall is very good for a small number of proposals/frame. However, it starts to saturate at about 40 proposals/frame, and for more than 90 proposals/frame the recall is considerably higher for the split-octave version. This is important for our application in which we have up to 43 objects visible per frame, so we chose the split-octave method for the remaining evaluations.

As external baselines, we chose some recent methods with available source code that have shown good performance for proposal detection, namely the objectness measure of Alexe et al. [ADF12], the Prime Object Proposals of Manén et al. [MGV13] (RP), and the contour detector with hierarchical image segmentation of Arbelaez et al. [AMFM11] (gPb). For all methods, we rank the proposals according to their quality (our saliency proposals: saliency measure; [ADF12]: objectness measure; [MGV13]: proposals are ranked already; [AMFM11] do not provide a score for their hierarchical regions. We extract regions with a watershed algorithm and use the difference between the maximum and the minimum contour score as region score). Since [ADF12] and [MGV13] deliver bounding boxes instead of pixel-precise regions, we use the smallest surrounding rectangle around the regions also for [AMFM11] and our own method to make the methods comparable. Fig. 4 shows that our object proposals have consistently a higher precision and recall than all the other methods.

Fig. 6: Correct tracking proposals (initialized with 200 saliency proposals).

In our video scenario, it is not only of interest how many objects are detected on a frame level, but even more how many objects of the real-world are found over the whole sequence. Some objects might not be found in one frame, but in another one. Therefore, we additionally measure the global recall,*i.e.*, the percentage of discovered objects from all objects that are visible over the entire sequence. The global recall is plotted over time to show how the detection rate develops. Since not all objects are visible in each frame, there is a maximal limit of recall that can be achieved. We plot this curve as a theoretical upper bound (in yellow). We evaluated all methods when considering 50, 100, or 200 proposals per frame. For better visibility, we plot all these curves only for one sequence and only show the 200 proposals/frame version for the other plots (complete plots in appendix). Since the sequences have a different number of frames, we consider them separately here and show them in Fig. 5. The plots show how the global recall evolves over time. When considering 200 proposals per frame, we are able to detect most objects over the whole sequence (between 75% (*coffee machine* seq.) and 96% recall (*kitchen C*)). With these values, our method outperforms all other methods (except seq. *kitchen C*, in which RP and our method achieve the same performance level). Note also that with only 100 proposals, we still detect the same number of objects as the second best method RP did with 200 proposals. The results in the other sequences are similar.

Tracking Proposals. We now evaluate the proposals generated by the tracker in comparison to the saliency proposals with which the tracker was initialized. Fig. 7 shows precision and recall over the number of proposals per frame. Our tracker is able to achieve a higher recall than the saliency proposals. The increased recall shows that the tracker is able to track good proposals into later frames, where they are not among the saliency proposals. In Fig. 8 we show the global recall over time compared to saliency proposals. Again, the tracker is able to achieve significantly higher recall. Fig. 6 shows the correct proposals made by the tracker in some example frames.

7 Conclusion

We have presented a new method for object proposal generation on a sequence-level. It is especially well suited for videos from mobile devices in which it is

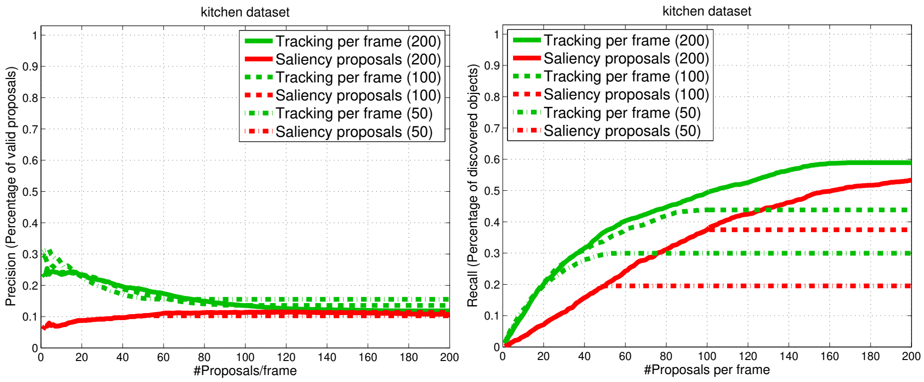

Fig. 7: Precision and recall per frame averaged over all sequences. Red: frame-level saliency proposals. Green: sequence level proposals.

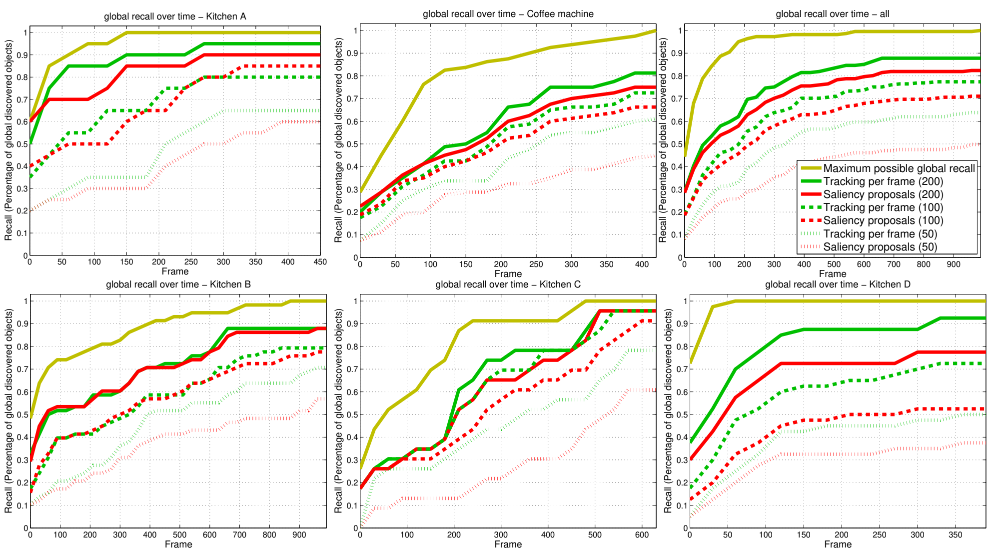

Fig. 8: Global recall over time measured for each individual sequence; the top right is the average of all sequences. Gold: maximum theoretical recall. Red: frame-level saliency proposals. Green: sequence level proposals.

important to limit the cost factor of recognition queries. Our method consists of two steps: first, a new frame-based object proposal detector based on multi-scale saliency determines proposals with a higher per-frame recall that current state-of-the-art methods. We show that this method is able to detect most of the objects in a scene even in very complex scenarios with plenty of objects and a high degree of clutter. Second, the proposals are tracked over time in order to group proposals that belong to the same real-world object and filter out inconsistent regions. Thus, our approach delivers a set of region trajectories that combine different views of an object. These two components result in a significant reduction of proposals compared to frame-based methods, while keeping a consistently high recall. We show that we are able to detect on average 88% of the objects with this method. Finally, we select a representative view for each track that can be used as a query for recognition in future work.

Acknowledgements This research has been funded by the DFG project "Situiertes Sehen zur Erfassung von Form und Affordanzen von Objekten" (FR 2598/5-1 and LE 2708/1-1) under the D-A-CH lead agency programme.

A Appendix: Complete Set of Plots

A.1 Precision/Recall (200 proposals)

Here, in Fig. 9 and 10, we show the plots that correspond to Fig. 4. While Fig. 4 showed the average values of all sequences, we show here the results for each sequence separately. As in Fig. 4, we measure the precision and the recall frame-wise. Hereby, precision is the percentage of valid proposals, recall the percentage of discovered objects in a frame. The red curve shows the saliency proposals, the green and pale blue approach are variants of our method that serve as internal baselines. The black, blue and pink curve denote the methods [MGV13] (RP), [ADF12] (objectness), and [AMFM11] (gPb) which are used as external baselines.

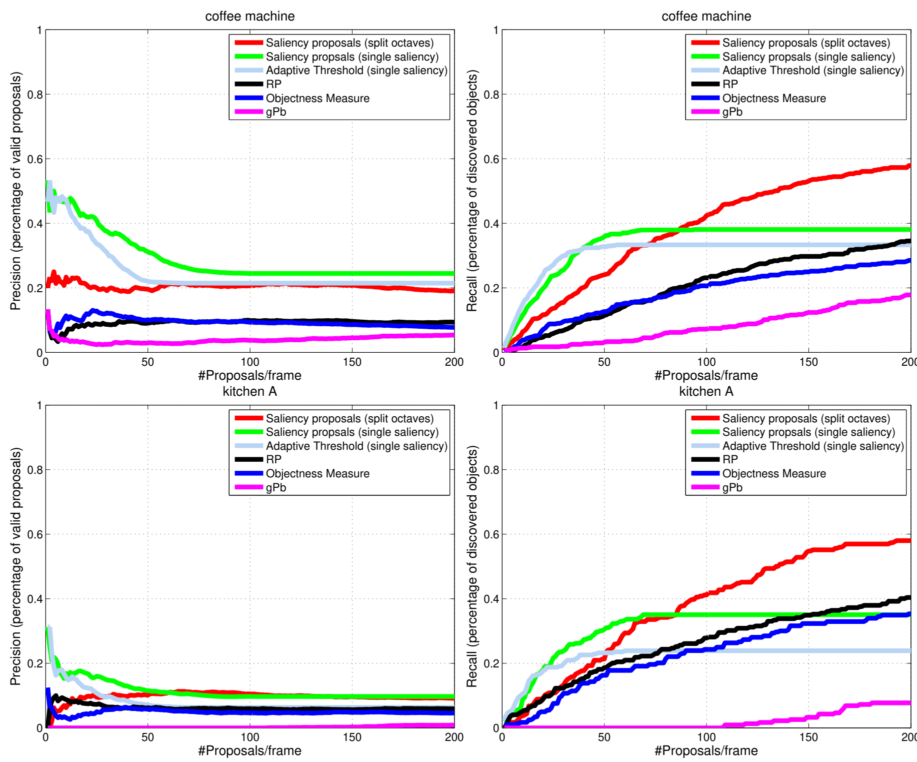

Fig. 9: Saliency proposals precision and recall for coffee machine and kitchen A.

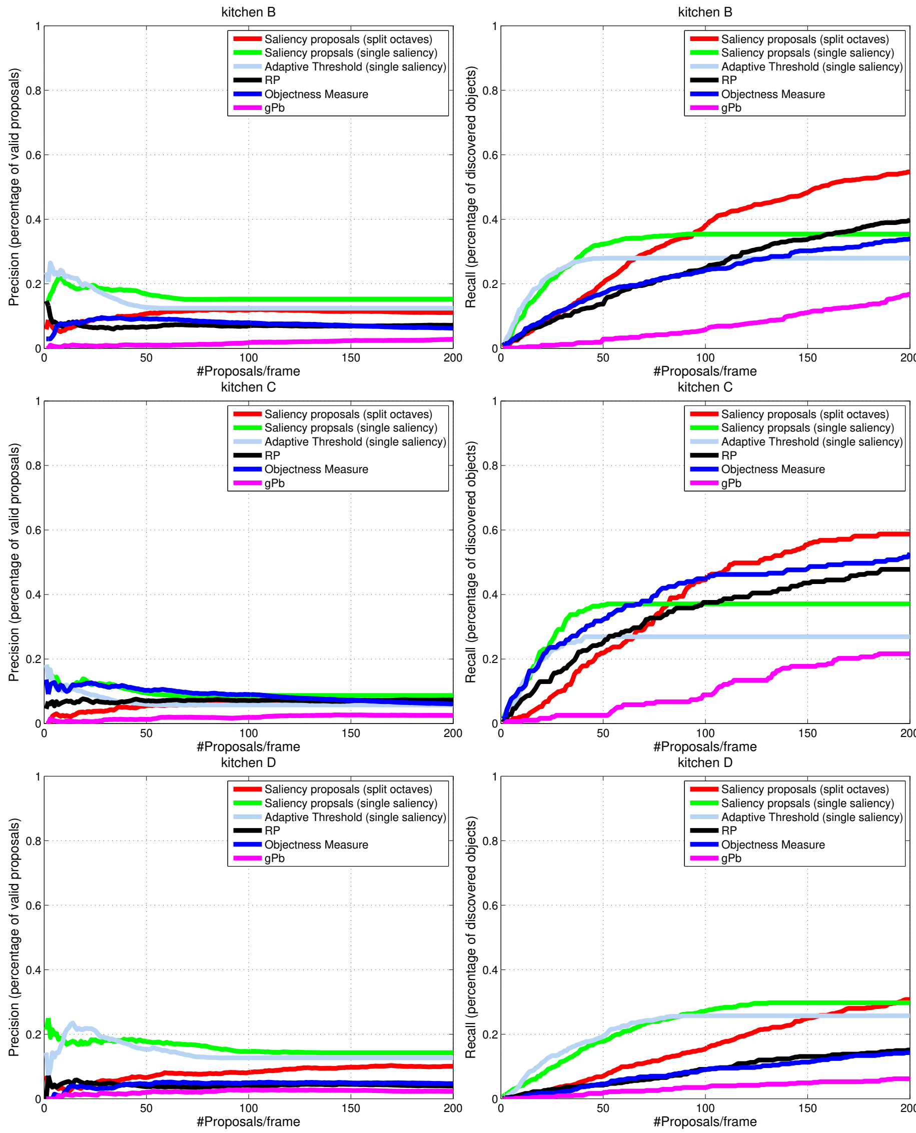

Fig. 10: Saliency proposals precision and recall for kitchen B, kitchen C, and kitchen D.

A.2 Global Recall Over Time

The plots in this section correspond to Fig. 5. They show the global recall over time, that means the percentage of real-world objects of the sequence that are have been detected at least once up to a specific point in time.

While above we showed only the 200-proposals-per-frame evaluation for better visibility (except for kitchen A), we show here additionally the evaluations for using 100 and 50 proposals per frame. Here again, the red curve is our saliency proposals, the other curves are the baseline methods.

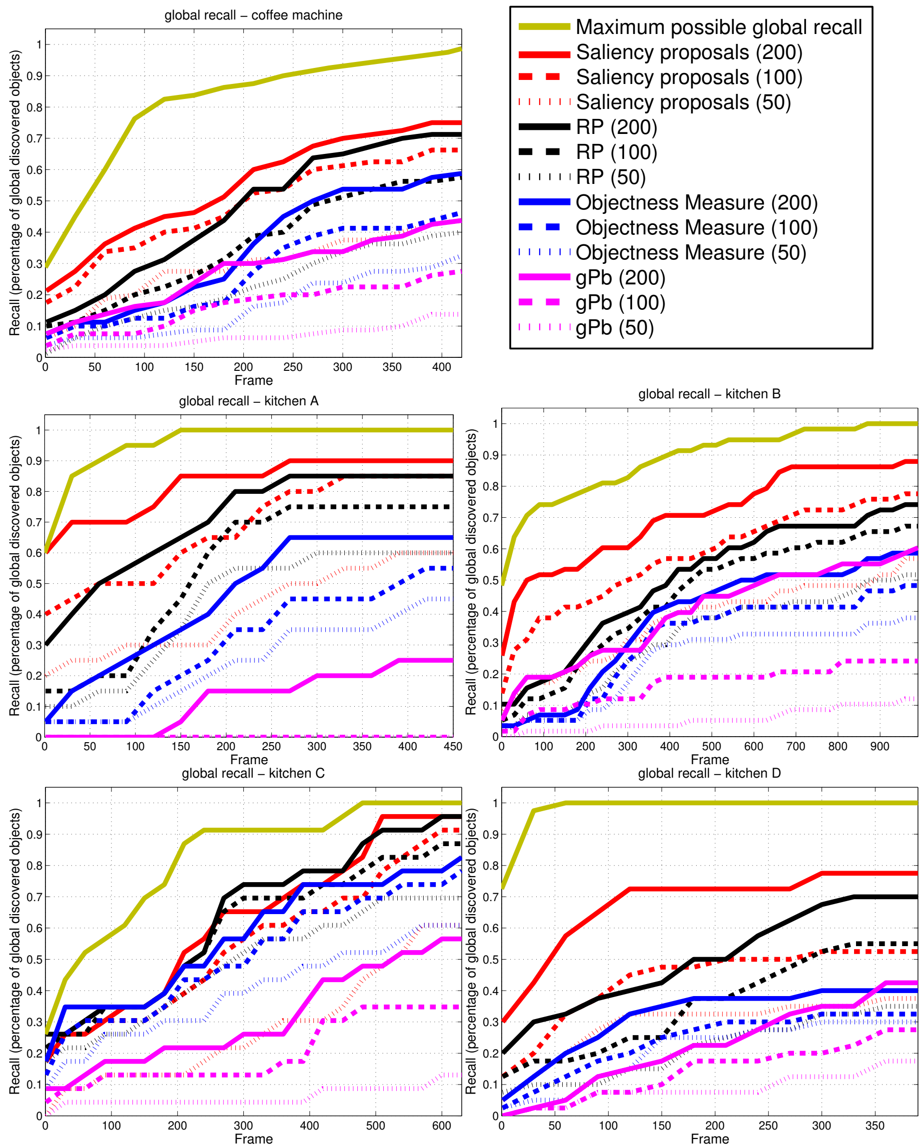

Fig. 11: Saliency proposals global recall over time.

A.3 Precision and Recall over Absolute Number of Proposals

In this section we show the precision and recall over the absolute number of proposals. With this evaluation we show how many individual objects were found in the whole sequence, regardless of which frame they were found in. For the saliency proposals this means we consider all proposals for every annotated frame. For the tracking proposals one proposals consists of one tracklet. Figures 12 to 16 show the plots, as well as the number of objects and number of frames for the five different sequences. These results show that with the tracking stage we can significantly reduce the number of proposals while still achieving higher precision and recall than with the saliency proposals. In all of the sequences we only need between 500 and 750 track-level proposals per sequence to reach above 80% recall – compared to the 200 saliency proposals per frame for each frame of the sequence that we start from, this is a significant reduction!

These plots should not be confused with the global recall over time, where the accumulated recall over the course of the video is shown.

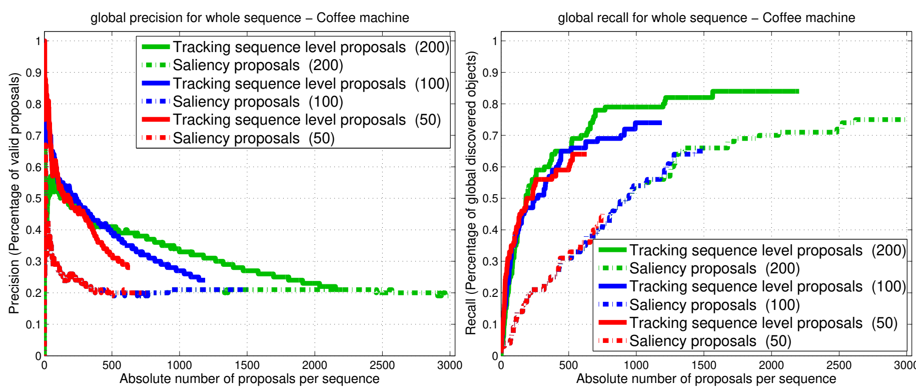

Fig. 12: Coffee machine: contains 80 individual objects in 15 annotated frames (437 frames total).

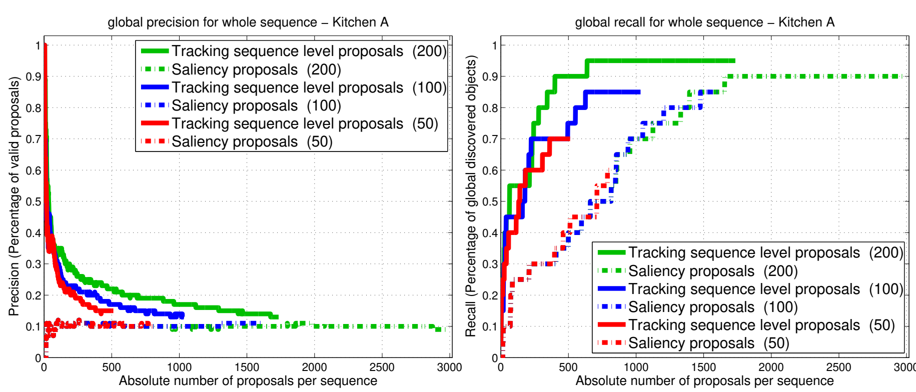

Fig. 13: Kitchen A: contains 20 individual objects in 16 annotated frames (479 frames total).

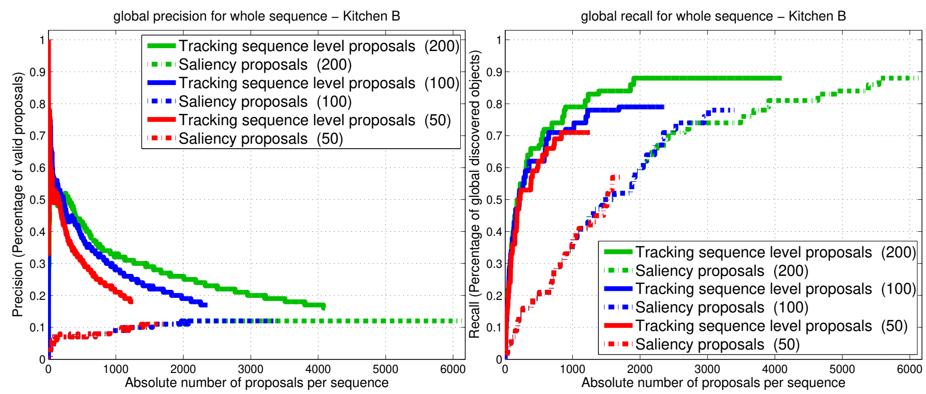

Fig. 14: Kitchen B: contains 58 individual objects in 34 annotated frames (1011 frames total).

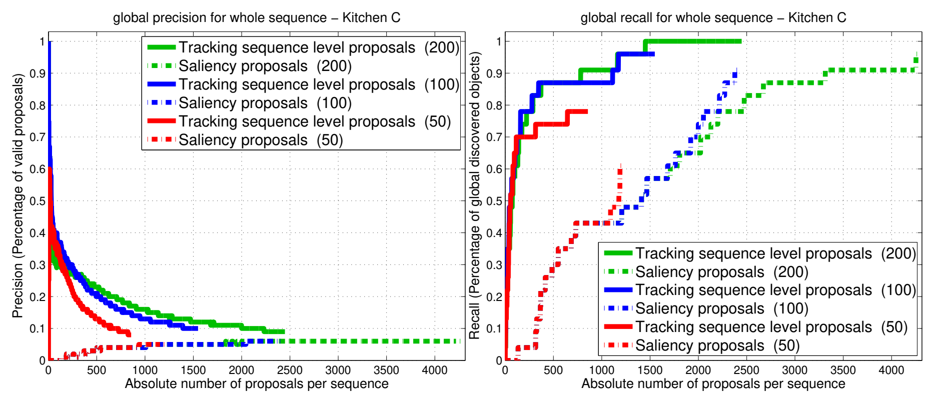

Fig. 15: Kitchen C: contains 23 individual objects in 24 annotated frames (704 frames total).

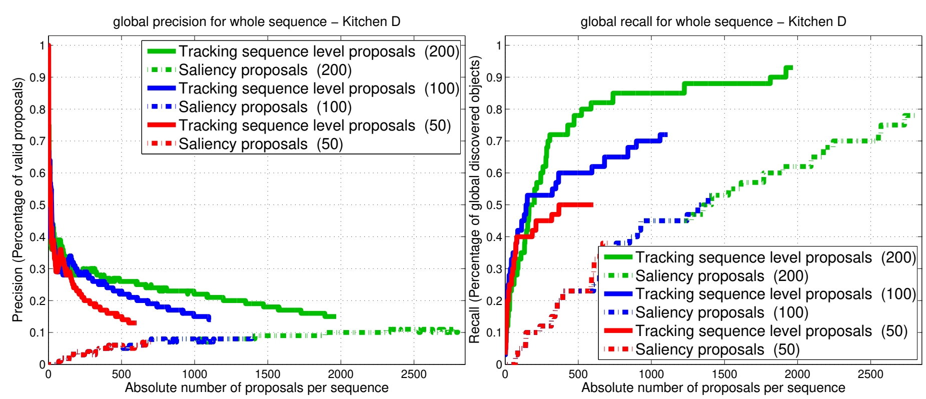

Fig. 16: Kitchen D: contains 40 individual objects in 14 annotated frames (406 frames total).

References

AB94. R. Adams and L. Bischof. Seeded Region Growing. *PAMI*, 16(6):641–647, 1994.

ADF12. B. Alexe, T. Deselaers, and V. Ferrari. Measuring the Objectness of Image Windows. *PAMI*, 34(11):2189–2202, 2012.

AMFM11. P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *PAMI*, 33(5):898–916, 2011.

AS10. R. Achanta and S. Süsstrunk. Saliency Detection using Maximum Symmetric Surround. In *ICIP*, 2010.

BBK13. M. Bjoerkman, N. Bergstroem, and D. Kragic. Detecting, Segmenting and Tracking Unknown Objects using MRF Inference. *CVIU*, 2013.

BI10. A. Borji and L. Itti. State-of-the-art in Visual Attention Modeling. *PAMI*, 2010.

BK10. M. Bjoerkman and D. Kragic. Active 3D Segmentation through Fixation of Previously Unseen Objects. In *BMVC*, 2010.

BR08. C. Bibby and I. Reid. Robust Real-Time Visual Tracking using Pixel-Wise Posteriors. In *ECCV*, 2008.

BR10. C. Bibby and I. Reid. Real-time Tracking of Multiple Occluding Objects using Level Sets. In *CVPR*, 2010.

BRR12. M. Bleyer, C. Rhemann, and C. Rother. Extracting 3D Scene Consistent Object Proposals and Depth from Stereo Images. In *ECCV*, 2012.

BT09. N. Bruce and J. Tsotsos. Saliency, Attention, and Visual Search: An Information Theoretic Approach. *Journal of Vision*, 9(3):1–24, 2009.

BTSI13. A. Borji, H.R. Tavakoli, D.N. Sihite, and L. Itti. Analysis of scores, datasets, and models in visual saliency modeling. In *ICCV*, 2013.

CS12. J. Carreira and C. Smichisescu. CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts. *PAMI*, 34(7):1312–1328, 2012.

DDS+09. J. Deng, W. Dong, R. Socher, L-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

DRS+13. T. Dean, M.A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. Fast, Accurate Detection of 100,000 Object Classes on a Single Machine. In *CVPR*, 2013.

EGW+10. M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 88(14):303–338, 2010.

EH14. I. Endres and D. Hoiem. Category-Independent Object Proposals with Diverse Ranking. *PAMI*, 2014.

FGMR10. P. Felzenszwalb, B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *PAMI*, 32(9), 2010.

FH04. P.F. Felzenszwalb and D.P. Huttenlocher. Efficient Graph-based Image Segmentation. *IJCV*, 59(2), 2004.

FMC14. S. Frintrop, G. Martín García, and A.B. Cremers. A Cognitive Approach for Object Discovery. In *International Conference on Pattern Recognition (ICPR) (accepted)*, 2014.

FSU13. S. Fidler, A. Sharma, and R. Urtasun. Bottom-Up Segmentation for Top-Down Detection. In *CVPR*, 2013.

HEH11. D. Hoiem, A.A. Efros, and M. Hebert. Recovering Occlusion Boundaries from an Image. *IJCV*, 91(3):328–346, 2011.

IKN98. L. Itti, C. Koch, and E. Niebur. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *PAMI*, 20(11):1254–1259, 1998.

KF11. D.A. Klein and S. Frintrop. Center-surround Divergence of Feature Statistics for Salient Object Detection. In *ICCV*, 2011.

KF12. D.A. Klein and S. Frintrop. Salient Pattern Detection using W_2 on Multivariate Normal Distributions. In *DAGM*, 2012. http://www.iai.uni-bonn.de/~kleind/.

KGF12. D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation Propagation in ImageNet. In *ECCV*, 2012.

KK13. G. Kootstra and D. Kragic. Fast and Bottom-Up Detection, Segmentation, and Evaluation using Gestalt Principles. In *ICRA*, 2013.

KMFF13. A. Karpathy, S. Miller, and L. Fei-Fei. Object Discovery in 3D Scenes via Shape Analysis. In *ICRA*, 2013.

KPB+05. R. Kaucic, A.G. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs. A Unified Framework for Tracking through Occlusions and Across Sensor Gaps. In *CVPR*, 2005.

KSH12. A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.
LG11. Y.J. Lee and K. Grauman. Learning the Easy Things First: Self-Paced Visual Category Discovery. In *CVPR*, 2011.
LYS+09. T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to Detect a Salient Object. *PAMI*, 2009.
MGV13. S. Manén, M. Guillaumin, and L. Van Gool. Prime Object Proposals with Randomized Prim's Algorithm. In *ICCV*, 2013.
MHEL10. D. Mitzel, E. Horbert, A. Ess, and B. Leibe. Multi-Person Tracking with Sparse Detection and Continuous Segmentation. In *ECCV*, 2010.
Pas97. H. Pashler. *The Psychology of Attention*. MIT Press, Cambridge, MA, 1997.
PKPH12. F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung. Saliency Filters: Contrast based Filtering for Salient Region Detection. In *CVPR*, 2012.
Ren00. R.A. Rensink. The Dynamic Representation of Scenes. *Visual Cognition*, 7:17–42, 2000.
Sch01. B.J. Scholl. Objects and Attention: the State of the Art. *Cognition*, 80:1–46, 2001.
STS11. J. Shin, R. Triebel, and R. Siegwart. Unsupervised 3D Object Discovery and Categorization for Mobile Robots. In *ISRR*, 2011.
SWLL13. K. Shi, K. Wang, J. Lu, and L. Lin. PISA: Pixelwise Image Saliency by Aggregating Complementary Appearance Contrast Measures with Spatial Priors. In *CVPR*, 2013.
UvGS13. J.R.R. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective Search for Object Recognition. *IJCV*, 104(2):154–171, 2013.
WPN12. D. Wang, I. Posner, and P. Newman. What Could Move? Finding Cars, Pedestrians and Bicyclists in 3D Laser Data. In *ICRA*, 2012.
YXSJ13. Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical Saliency Detection. In *CVPR*, 2013.
YZL+13. C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency Detection via Graph-based Manifold Ranking. In *CVPR*, 2013.

Aachener Informatik-Berichte

This list contains all technical reports published during the past three years. A complete list of reports dating back to 1987 is available from:

http://aib.informatik.rwth-aachen.de/

To obtain copies please consult the above URL or send your request to:

Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen,
Email: biblio@informatik.rwth-aachen.de

- 2011-01 * Fachgruppe Informatik: Jahresbericht 2011
- 2011-02 Marc Brockschmidt, Carsten Otto, Jürgen Giesl: Modular Termination Proofs of Recursive Java Bytecode Programs by Term Rewriting
- 2011-03 Lars Noschinski, Fabian Emmes, Jürgen Giesl: A Dependency Pair Framework for Innermost Complexity Analysis of Term Rewrite Systems
- 2011-04 Christina Jansen, Jonathan Heinen, Joost-Pieter Katoen, Thomas Noll: A Local Greibach Normal Form for Hyperedge Replacement Grammars
- 2011-06 Johannes Lotz, Klaus Leppkes, and Uwe Naumann: dco/c++ - Derivative Code by Overloading in C++
- 2011-07 Shahar Maoz, Jan Oliver Ringert, Bernhard Rumpe: An Operational Semantics for Activity Diagrams using SMV
- 2011-08 Thomas Ströder, Fabian Emmes, Peter Schneider-Kamp, Jürgen Giesl, Carsten Fuhs: A Linear Operational Semantics for Termination and Complexity Analysis of ISO Prolog
- 2011-09 Markus Beckers, Johannes Lotz, Viktor Mosenkis, Uwe Naumann (Editors): Fifth SIAM Workshop on Combinatorial Scientific Computing
- 2011-10 Markus Beckers, Viktor Mosenkis, Michael Maier, Uwe Naumann: Adjoint Subgradient Calculation for McCormick Relaxations
- 2011-11 Nils Jansen, Erika Ábrahám, Jens Katelaan, Ralf Wimmer, Joost-Pieter Katoen, Bernd Becker: Hierarchical Counterexamples for Discrete-Time Markov Chains
- 2011-12 Ingo Felscher, Wolfgang Thomas: On Compositional Failure Detection in Structured Transition Systems
- 2011-13 Michael Förster, Uwe Naumann, Jean Utke: Toward Adjoint OpenMP
- 2011-14 Daniel Neider, Roman Rabinovich, Martin Zimmermann: Solving Muller Games via Safety Games
- 2011-16 Niloofar Safiran, Uwe Naumann: Toward Adjoint OpenFOAM
- 2011-17 Carsten Fuhs: SAT Encodings: From Constraint-Based Termination Analysis to Circuit Synthesis
- 2011-18 Kamal Barakat: Introducing Timers to pi-Calculus
- 2011-19 Marc Brockschmidt, Thomas Ströder, Carsten Otto, Jürgen Giesl: Automated Detection of Non-Termination and NullPointerExceptions for Java Bytecode
- 2011-24 Callum Corbett, Uwe Naumann, Alexander Mitsos: Demonstration of a Branch-and-Bound Algorithm for Global Optimization using McCormick Relaxations

2011-25 Callum Corbett, Michael Maier, Markus Beckers, Uwe Naumann, Amin Ghobeity, Alexander Mitsos: Compiler-Generated Subgradient Code for McCormick Relaxations

2011-26 Hongfei Fu: The Complexity of Deciding a Behavioural Pseudometric on Probabilistic Automata

2012-01 Fachgruppe Informatik: Annual Report 2012

2012-02 Thomas Heer: Controlling Development Processes

2012-03 Arne Haber, Jan Oliver Ringert, Bernhard Rumpe: MontiArc - Architectural Modeling of Interactive Distributed and Cyber-Physical Systems

2012-04 Marcus Gelderie: Strategy Machines and their Complexity

2012-05 Thomas Ströder, Fabian Emmes, Jürgen Giesl, Peter Schneider-Kamp, and Carsten Fuhs: Automated Complexity Analysis for Prolog by Term Rewriting

2012-06 Marc Brockschmidt, Richard Musiol, Carsten Otto, Jürgen Giesl: Automated Termination Proofs for Java Programs with Cyclic Data

2012-07 André Egners, Björn Marschollek, and Ulrike Meyer: Hackers in Your Pocket: A Survey of Smartphone Security Across Platforms

2012-08 Hongfei Fu: Computing Game Metrics on Markov Decision Processes

2012-09 Dennis Guck, Tingting Han, Joost-Pieter Katoen, and Martin R. Neuhäußer: Quantitative Timed Analysis of Interactive Markov Chains

2012-10 Uwe Naumann and Johannes Lotz: Algorithmic Differentiation of Numerical Methods: Tangent-Linear and Adjoint Direct Solvers for Systems of Linear Equations

2012-12 Jürgen Giesl, Thomas Ströder, Peter Schneider-Kamp, Fabian Emmes, and Carsten Fuhs: Symbolic Evaluation Graphs and Term Rewriting — A General Methodology for Analyzing Logic Programs

2012-15 Uwe Naumann, Johannes Lotz, Klaus Leppkes, and Markus Towara: Algorithmic Differentiation of Numerical Methods: Tangent-Linear and Adjoint Solvers for Systems of Nonlinear Equations

2012-16 Georg Neugebauer and Ulrike Meyer: SMC-MuSe: A Framework for Secure Multi-Party Computation on MultiSets

2012-17 Viet Yen Nguyen: Trustworthy Spacecraft Design Using Formal Methods

2013-01 * Fachgruppe Informatik: Annual Report 2013

2013-02 Michael Reke: Modellbasierte Entwicklung automobiler Steuerungssysteme in Klein- und mittelständischen Unternehmen

2013-03 Markus Towara and Uwe Naumann: A Discrete Adjoint Model for OpenFOAM

2013-04 Max Sagebaum, Nicolas R. Gauger, Uwe Naumann, Johannes Lotz, and Klaus Leppkes: Algorithmic Differentiation of a Complex C++ Code with Underlying Libraries

2013-05 Andreas Rausch and Marc Sihling: Software & Systems Engineering Essentials 2013

2013-06 Marc Brockschmidt, Byron Cook, and Carsten Fuhs: Better termination proving through cooperation

2013-07 André Stollenwerk: Ein modellbasiertes Sicherheitskonzept für die extrakorporale Lungenunterstützung

2013-08 Sebastian Junges, Ulrich Loup, Florian Corzilius and Erika Ábrahám: On Gröbner Bases in the Context of Satisfiability-Modulo-Theories Solving over the Real Numbers

2013-10 Joost-Pieter Katoen, Thomas Noll, Thomas Santen, Dirk Seifert, and Hao Wu: Performance Analysis of Computing Servers using Stochastic Petri Nets and Markov Automata

2013-12 Marc Brockschmidt, Fabian Emmes, Stephan Falke, Carsten Fuhs, and Jürgen Giesl: Alternating Runtime and Size Complexity Analysis of Integer Programs

2013-13 Michael Eggert, Roger Häußling, Martin Henze, Lars Hermerschmidt, René Hummen, Daniel Kerpen, Antonio Navarro Pérez, Bernhard Rumpe, Dirk Thißen, and Klaus Wehrle: SensorCloud: Towards the Interdisciplinary Development of a Trustworthy Platform for Globally Interconnected Sensors and Actuators

2013-14 Jörg Brauer: Automatic Abstraction for Bit-Vectors using Decision Procedures

2013-19 Florian Schmidt, David Orlea, and Klaus Wehrle: Support for error tolerance in the Real-Time Transport Protocol

2013-20 Jacob Palczynski: Time-Continuous Behaviour Comparison Based on Abstract Models

2014-01 * Fachgruppe Informatik: Annual Report 2014

2014-04 Namit Chaturvedi: Languages of Infinite Traces and Deterministic Asynchronous Automata

2014-05 Thomas Ströder, Jürgen Giesl, Marc Brockschmidt, Florian Frohn, Carsten Fuhs, Jera Hensel, and Peter Schneider-Kamp: Automated Termination Analysis for Programs with Pointer Arithmetic

* These reports are only available as a printed version.

Please contact `biblio@informatik.rwth-aachen.de` to obtain copies.