To appear, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

Boundary Preserving Dense Local Regions

Jaechul Kim and Kristen Grauman
University of Texas at Austin
{jaechul,grauman}@cs.utexas.edu

Abstract

We propose a dense local region detector to extract features suitable for image matching and object recognition tasks. Whereas traditional local interest operators rely on repeatable structures that often cross object boundaries (e.g., corners, scale-space blobs), our sampling strategy is driven by segmentation, and thus preserves object boundaries and shape. At the same time, whereas existing region-based representations are sensitive to segmentation parameters and object deformations, our novel approach to robustly sample dense sites and determine their connectivity offers better repeatability. In extensive experiments, we find that the proposed region detector provides significantly better repeatability and localization accuracy for object matching compared to an array of existing detectors. In addition, we show our regions lead to excellent results on two benchmark tasks that require good feature matching: weakly supervised foreground discovery, and nearest neighbor-based object recognition.

1. Introduction

Local features are a basic building block for image retrieval and recognition tasks. Their locality offers robustness to occlusions and deformation, and when extracted densely and/or at multiple scales they capture rich statistics for recognition algorithms (e.g., for a bag of words representation). The general local feature pipeline consists of (a) a *detection* stage, which selects the image sites (positions, scales, shapes) where features will be extracted, and (b) a *description* stage, which uses the image content at each such site to form a local descriptor. This work is concerned with the detection stage.

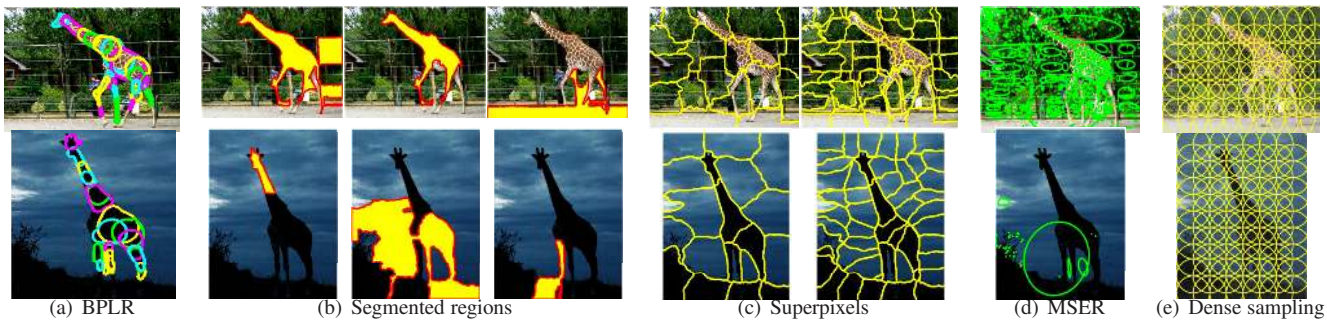
Researchers have developed a variety of techniques to perform detection, ranging from sophisticated interest point operators [19, 20, 21, 12, 31] to dense sampling strategies [23]. While by design such methods provide highly repeatable detections across images, their low-level local sampling criteria generate many descriptors that straddle object boundaries, and—if they are too local—may also lack distinctiveness (i.e., patches of texture vs. actual object parts). On the other hand, while segmentation algorithms can produce boundary-preserving base features and reveal object shape [26, 11, 30, 18], they tend to be sensitive to global image variations and so lack repeatability.

Figure 1. Boundary-preserving local regions (BPLRs) capture local object shape with dense spatial coverage. (We densely extract BPLRs across the image, but for visualization purposes this figure displays only a few.)

Our goal is to address this current tradeoff, and create a detector for features that are both distinctive *within* the image as well as repeatable *across* images. To this end, we propose a novel dense local region extraction algorithm driven by segmentation.

Briefly, it works as follows: given multiple overlapping segmentations of the input image, we first compute their corresponding distance transform maps. We then divide each segment into regular grid cells, and sample an "element" feature in each cell, whose position and associated scale are determined by the maximal distance transform value in the cell. This step yields elements that avoid overlapping object boundaries, and tend to be closest to other elements within the same segment. Next we link all elements with a minimum spanning tree, which extends connections beyond the original segment boundaries and aptly integrates the multiple segmentations. Finally, we extract a dense set of overlapping regions, each of which consists of a group of linked (neighboring) elements within the tree. We call the resulting regions *boundary-preserving local regions* (BPLRs). See Figure 1.

Because our extracted regions tend to preserve object boundaries, they are informative for object shape. At the same time, because they link sampled elements across multiple segmentations, they are robust to unstable segmentations and thus repeatable across images. Finally, their dense coverage of the image ensures we retain reliable feature statistics that are critical for recognition and matching.



(a) BPLR (b) Segmented regions (c) Superpixels (d) MSER (e) Dense sampling

Figure 2. Illustration of BPLR's key contrasts with representative existing detectors. **(a) The proposed BPLR** features are reliably repeated across different object instances in spite of large intra-class variation in pose and appearance. They respect object boundaries while maintaining good spatial coverage per region. (Note, we display only a sample for different fg object parts; our complete extraction is dense and covers entire image.) **(b) Regions** from a segmentation algorithm (here, obtained with [2], and pruned to only foreground-overlapping regions) typically produce some high quality segments, but the shape and localization often lacks repeatability across instances. Further, if a good segment encompasses the entire object, it won't match other instances with deformation. **(c) Superpixels** (obtained here with Normalized Cuts) are also local and dense, but typically lose informative shape cues and lack repeatability (compare shapes of superpixels on the two giraffe instances). **(d) Local interest regions** (obtained with MSER [19]) are highly repeatable for multiple views of the same instance, but do not respect object boundaries and fire very differently across different instances of the same object class. **(e) Dense patches** offer good coverage and "brute force" repeatability, but many features straddle object boundaries, and shape is mostly not preserved.

We evaluate our BPLR detector's repeatability (how well foreground features on an object match others in the same class) and localization accuracy (how accurately feature matches can predict objects' positions and scales) with extensive experiments on benchmark datasets. Direct comparisons to several existing extractors—interest regions, dense local patches, semi-local feature configurations, and segments—show its clear advantages, particularly for deformable objects and those with characteristic shape. Finally, having examined its quality as a raw detector, we employ the BPLR within two higher-level applications that require good feature matching: foreground segmentation and nearest-neighbor object classification. Our detector offers significant gains relative to alternative extraction methods and improves upon the state-of-the-art.

2. Background and Related Work

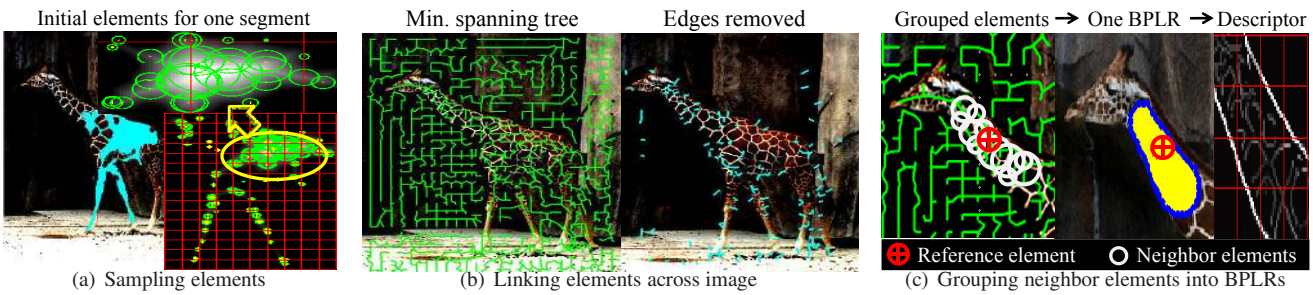
We now review related work on feature detection; Figure 2 depicts the key contrasts to our approach.

Local interest region detection is a long-standing research topic in computer vision, and scale or affine-invariant local regions [19, 20, 21, 12] are critically valuable for multi-view matching problems like wide-baseline stereo or instance recognition. For generic object categories, on the other hand, they tend to be too sparse; densely sampled local patches offer better coverage and are regularly found to outperform interest points (e.g., see [23]), at the cost of much greater storage and computation. Recent work on dense interest points [31] shows how to merge advantages of either sampling strategy, balancing coverage with repeatability. Due to their inherent locality, however, individual features from any such detector can lack distinctiveness, and will rarely fire on a true "part" of an object (e.g., a giraffe's neck).

One way to enhance distinctiveness is to group nearby local features into neighborhoods or "semi-local" configurations, exploiting geometric consistency observed across training instances [25, 8, 15]. Strong inter-feature geometric constraints can be too restrictive (non-repeatable) for generic objects, whereas grouping methods that require class-specific supervision are not applicable to bottom-up processing of arbitrary images. Instead, we propose a grouping stage that links element features according to region and contour structures throughout the image, and assume neither rigid layouts nor class-specific supervision.

Due to steady advances in bottom-up segmentation algorithms [2], increasingly researchers are considering how to employ *segments* as base features, in place of local patches [11, 30, 24, 26]. Segments are appealing since they capture object shape and have broader spatial coverage. However, the instability of segmentation algorithms with respect to image variations can make the features' shapes unreliable or sensitive to parameter settings. Thus, existing work often focuses on how to select reliable segment-parts using labeled data [11, 30]. Multiple segmentations (generated by varying the segmentation parameters) are often used to expand the pool of candidates for a single image (e.g., [18, 10]). Whereas existing methods typically try to find "good" full-object segments among this pool, we show how to incorporate all segmentation hypotheses when both sampling and linking the element features.

Much less attention has been given to the interplay between low-level local features and segmentation. The segmentation-based interest points proposed in [14] consist of ellipses fit to segment areas and corners computed on segment boundaries. In contrast to our approach, however, corners may often miss shape cues of the regions, and fitting ellipses directly to segments can be susceptible to seg-



(a) Sampling elements (b) Linking elements across image (c) Grouping neighbor elements into BPLRs

Figure 3. Main components of the approach. Best viewed in color. **(a)** For each initial segment, we sample local elements densely in a grid according to its distance transform (left: segment; lower right: grid; upper right: zoom-in to show sampled elements and their scales). **(b)** Elements are linked across the image, using the overlapping multiple segmentations to create a single structure that reflects the main shapes and segment layout. **(c)** Using that structure, we extract one BPLR per element. Each BPLR is a group of neighboring elements. Finally, the BPLR is mapped to some descriptor (we use PHOG+gPb).

mentation errors. An interesting approach proposed in [16] groups superpixels into objects' skeletal parts. Their use of medial axis points is related to our use of the distance transform; however, we seek dense and generic local features rather than only symmetric parts.

Please note that our work and those cited above all tackle region *detection*; we use existing descriptors to capture our detected regions' shape, and standard matching techniques to demonstrate their applicability. Thus, work on shape descriptors and contour matching (e.g., [3, 9]) is complementary but separate from our focus.

3. Approach

We first describe how we sample initial elements using the input segmentations (Sec. 3.1). Then we explain how to link these elements across the image (Sec. 3.2). Finally, we show how to use the computed structure to extract dense groups of elements, each of which is a shape-preserving region (Sec. 3.3).

3.1. Sampling Initial Elements

Given an image, we first obtain multiple overlapping segmentations. (We use the state-of-the-art algorithm developed in [2] to produce a high quality hierarchy of segments, though other methods are possible.) These segmentation hypotheses do not serve as detected regions; rather, we use them to guide the extraction of initial component features that we call "elements". Each element is a circle with a position (its center) and associated scale (its radius).

The goal of our novel sampling strategy is to balance both density and object boundary preservation. To that end, we compute a distance transform (DT) from the boundary edges of each segment, and then subdivide the segment into a dense grid of cells (e.g., 4×4 pixels per cell). For each cell, we sample an element at the location with the maximal distance transform value within the cell, and set the radius of the element by that maximal distance value. Figure 3(a) shows sampled elements from one segment.

Selecting elements' scale by the DT prevents them from overlapping the originating segment's boundary. At the same time, refining the dense sampling positions by the maximal DT values pushes sampled locations to the inner part of each segment, keeping elements originating from the same segment closer to one another than those from different segments. Due to this geometric property, when we link elements across all segments in the next stage (Sec. 3.2), we have a soft preference to join elements originating from the same segment. In addition, the local nature of our sampling approach limits the influence of segment "errors"; that is, holes or leaks (relative to the true object boundaries) do not destroy the sampling and scale selection. Thus, we retain a large number of good elements that respect object boundaries even with partially flawed segments.

3.2. Linking Elements Throughout the Image

Next we want to take these elements and define the neighborhood structure across the entire image, which in turn will determine how we extract groups of neighboring elements to form BPLRs. A naive linking of the elements based on their spatial (image) distance would fail to capture the image-wide contours and shape revealed by the multiple segmentation hypotheses. Instead, we define a two-step linking procedure that accounts for this structure and reduces cross-object connections.

The first step computes a global linkage graph connecting all element locations via a minimum spanning tree, where each edge weight is given by the Euclidean distance between the two points it connects. By minimizing the sum of total edge weights, the resulting spanning tree removes the longer edges from the graph—most of which cross object boundaries due to the geometric property of the DT-based sampling. As a result, we have a global link structure respecting object boundaries, in which every element has at least one direct neighbor (see Figure 3(b), left image).

Whereas the above step reduces connectivity for more distant elements, we also want to reduce connectivity for

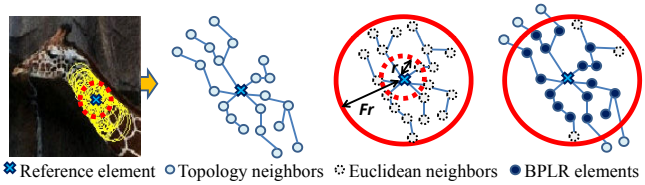


Figure 4. Grouping neighboring elements relative to a reference element. Topology neighbors: up to $N(= 3)$ hops for the reference; Euclidean neighbors: within F times the scale of the reference; BPLR elements: intersection of topology and Euclidean neighbors.

elements divided by any apparent object contours. Thus, in the second linkage step, we compute a simple post-processing of the spanning tree that removes noisy tree edges that cross strong intervening contours. We compute the contour strength at each pixel using the "globalized probability of boundary" (gPb) detector [17], and remove links crossing contours exceeding the average non-zero gPb value in the image. Figure 3(b) (right image) shows the types of links removed by this stage; we see that most do indeed cross object boundaries. Nonetheless, even an erroneous pruning at this stage has limited impact, given the density of the elements and the manner in which we ultimately group them into regions, as we explain in the next section.

3.3. Grouping Neighboring Elements into Regions

Finally, we use the elements and the computed graph to extract a dense set of boundary-preserving local regions (BPLRs). For every element (i.e., every node in the graph), we create one BPLR. Each BPLR consists of that "reference" element, plus a group of its neighbors in the graph (see Figure 3(c)).

We define the neighborhood based on two measures: topological distance in the graph (how many link hops separate the elements), and Euclidean distance in the image (L_2 distance between the elements' centers). The neighbors for a reference element are those within the intersection of regions spanned by either distance. Specifically, the topological neighborhood consists of any elements within N hops along the graph relative to the reference element, while the Euclidean neighborhood consists of any elements within a radius equal to F times the reference element's scale r (see Figure 4). Note that the topological radius is fixed over all elements in the graph (and all images), while the Euclidean radius is proportional to each element's scale.

Why the two distances? Using the Euclidean distance alone would maintain scale invariance, but is blind to the graph connectivity, which intentionally accounts for estimated image boundaries. On the other hand, topological distance accounts for this connectivity, and in the face of unstable segmentations, it tends to select neighbors better than the elements' noisy scale estimates; but, if used alone, it would not be robust to significant scale changes. Thus, our design is intended to balance the good parts of both.

The neighbors of each reference element within this intersected area form a BPLR. Since we extract the BPLRs for every densely sampled element, the resulting detections are also dense. The exact number per image depends on the initial segmentation and sampling grid; to give a concrete sense, using the Berkeley segmentation code we obtain about 150-250 segments, and then our method generates ∼7,000 features per image. To extract multi-scale features, we run BPLR detection on an image pyramid.[1]

While earlier uses of the distance transform for shape-based representations require fairly clean segmentation (e.g., a pure silhouette for medial axis or shock graph extraction [29]), our scheme remains quite robust with challenging natural images due to its linking procedure and dense sampling. By definition our approach has some dependence on the original set of multiple segmentations; however, because our linking scheme connects elements *beyond* their originating segment, it is fairly robust to segmentation variations, recovering larger descriptive regions that partially overlap different segments. In general, we'd prefer the input err towards finer segments, since we will produce candidate regions that join them.

Our approach performs region detection. To use these regions for matching, we need to further extract a *descriptor* for every region. One could employ any descriptor with our detector. In our experiments, we use Pyramids of Histograms of Oriented Gradients (PHOG) [6] computed over the gPb-edge map (see right image in Fig. 3(c)), which is similar to the descriptor used in [11]. It represents the outline of the shape as well as (coarsely) its inner texture, and thus is a good match for BPLR's strengths. To extract the PHOG+gPb feature, we put a bounding box around the BPLR, and nullify gPb values outside of the BPLR boundaries, excluding external edges from the histogram counts.

4. Results

The main goals of the experiments are 1) to demonstrate the raw quality of our region detector, and 2) to show its effectiveness when used for tasks that require reliable feature matching. For the first aspect, we analyze repeatability and localization accuracy across object categories (Sec. 4.1 and 4.2). For the second, we apply BPLR to foreground discovery and object classification (Sec. 4.3 and 4.4).

Datasets: We use four public datasets: the ETHZ Shape Classes [9], the ETH-TUD set collated by [25], the Caltech-28 set collated by [7], and the Caltech-101.

Implementation details: We generate multiple overlapping segmentations for each image using the algorithm

[1] One could alternatively take neighborhoods of multiple topological N hops and Euclidean F scales, although we did not observe any advantage over the image pyramid approach in practice.

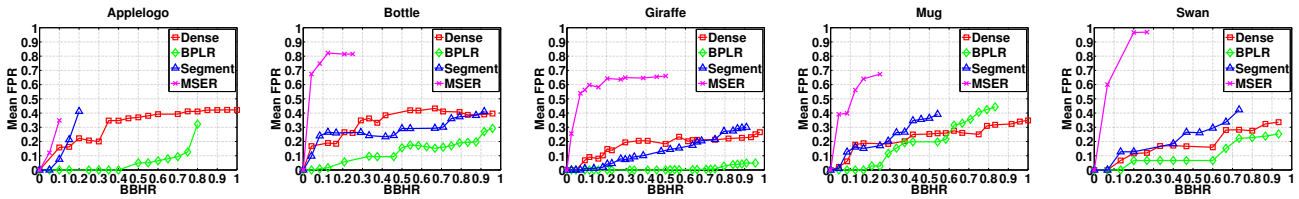


Figure 5. Repeatability on ETHZ objects. Plots compare our approach (BPLR) to three alternative region detectors: MSER, dense sampling, and segments. Quality is measured by the bounding box hit rate-false positive rate tradeoff (BBHR-FPR). Curves that are lower on the y-axis (fewer false positives) and longer along the x-axis (higher hit rate) are better.

of [2], with the authors' publicly available code. We vary parameters so as to provide 20-200 segments per segmentation, pool all the segments, and use them as input to our algorithm throughout. We extract BPLRs from elements sampled in grid cells of 4×4 pixels with $F = 2.5$, $N = 25$. To link elements in the minimum spanning tree, we use code by [28]. This setting generates on average 6,000-8,000 BPLRs in a 400×300 image, and takes about 70-90 seconds on a machine with a 3.4GHz CPU. Most of the time is spent computing topological distances and intervening contour strength among pairs of graph nodes; run-time drops quickly for sparser samplings (e.g., 2-5 seconds for 1,000-1,500 BPLRs). For each BPLR we create a PHOG+gPb descriptor using 3 pyramid levels (i.e., up to 4×4 subwindows) and 8 orientation bins, for a 168-dimensional descriptor. To "match" features, we simply use nearest neighbor (NN) search with Euclidean distance on the descriptors; for efficiency, we use [22].

Baselines: We compare to several state-of-the-art results in the literature ([25, 15, 1, 7] and many Caltech-101 numbers), plus three alternative extraction methods: 1) **MSER+SIFT:** MSER is the best local interest region in the evaluation by [21]; we use the Oxford code to generate 1000-1500 MSERs per image, 2) **Dense+SIFT:** sampled at a regular grid every 4 pixels in the image, over 5 scales of an image pyramid, and 3) **Segment+PHOG:** the same overlapping segments that serve as input to our algorithm, coupled with the same PHOG+gPb descriptor. Note, the former two baselines are widely used in the recognition literature, while the last is used in the state-of-the-art region-based approach of [11], making these strong and very informative baselines.

4.1. Repeatability for Object Categories

When matching images of the *same* scene or object, one can test repeatability by synthetically warping the images with parametric transformations (e.g., see [21]). However, such measures are not applicable to images of *generic objects*, where the goal is to ensure similar object parts are detected across instances. Thus, we quantify repeatability using the *Bounding Box Hit Rate - False Positive Rate* (BBHR-FPR) metric defined in [25]. To compute the BBHR-FPR, one selects features in the cluttered test image that have a match distance below a threshold with foreground features in the training images,[2] and declare a "hit" if at least five such features are inside the test image's bounding box. FPR counts those selected test features outside its bounding box. Sweeping through all distance thresholds, one records this average hit rate and corresponding FPR for all test images to form a BBHR-FPR curve. In short, the metric captures to what extent the selected features are repeatedly detected on the object foregrounds.

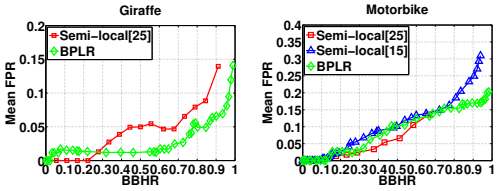


Figure 6. Repeatability on ETH+TUD objects. Plots compare our approach (BPLR) to two state-of-the-art semi-local feature methods [25, 15]. ([15] does not report results on the Giraffe class.)

Figure 5 shows the results for the ETHZ Shape Classes dataset, using a 50-50 train-test split. Our BPLR outperforms all the baselines. The BBHR is boosted by the density of our features, yet still maintains a low false positive rate. This indicates that BPLRs are highly repeatable across these shape-based categories, reliably discerning object foreground from background. In particular, we see that BPLR has the greatest advantage on the Giraffe class (center plot); this supports our claim that our shape-preserving dense local regions are better for handling deformable objects, given the giraffes' variable articulated poses.

Figure 6 compares to two state-of-the-art semi-local feature extraction methods [25, 15], using the ETH+TUD data and setup defined in [25].[3] Both previous methods build configurations of neighboring visual words, making them relevant to our approach to group element features. Our BPLR outperforms both. Again, gains on the non-rigid objects emphasize BPLR's strength for shape-based objects.

[2]Features in the training images are labeled as foreground when they are inside the bounding box and their best match is inside another training bounding box. The second condition reduces the ambiguity of bounding box annotation, e.g., background grass in a giraffe's bounding box.

[3]We exclude the Bike class, since it contains duplicated images in the test and training set, which inflates our results significantly.

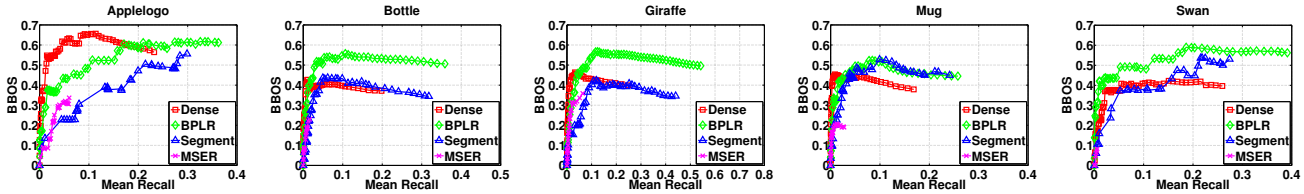


Figure 7. Localization accuracy on ETHZ objects. Plots compare our approach (BPLR) to three alternative region detectors: MSER, dense sampling, and segments. Quality is measured by the bounding box overlap score - recall (BBOS-Recall), which captures the layout of the feature matches. Curves that are higher in the y-axis (better object overlap) and longer along the x-axis (higher recall) are better.

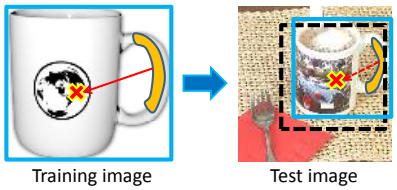


Figure 8. Given two matched regions and their relative scales, we project the training exemplar's bounding box into the test image (dotted rectangle). That match's BBOS is the overlap ratio between the projected box and the object's true bounding box.

4.2. Localization Accuracy

The BBHR-FPR reveals repeatability, but not layout. Ideally, the detected regions would also match with spatial consistency; i.e., if a region is detected on the fender of the car in one image, we want the fender on a different car in another image to also be detected, with a similar shape.

To quantify this, we introduce the *Bounding Box Overlapping Score - Recall* (BBOS-Recall) metric. For each feature in a test image, we match it to the training features, and use each match's position and scale to project the training example's bbox into the test image. The BBOS is the ratio between the intersection and union of this projected box and the test image's ground truth (see Figure 8). The recall is the portion of foreground test features that match a training foreground feature; false matches (to background) affect recall but not BBOS. A BBOS-Recall curve sweeps through all distance thresholds, and records the average BBOS and recall over all test images. In short, the metric captures the features' distinctiveness and localization accuracy.

Figure 7 shows the result for the ETHZ Shape data. In four of the five classes, our approach outperforms all the baselines, showing that its boundary-preserving quality helps localization. It is particularly strong for the shape-varying classes, Giraffe and Swan. In contrast, sparse MSERs—while highly repeatable for matching the same object—are poorly repeatable under intra-class variations, showing the lowest recall among the baselines. In addition, uniformly-shaped dense patches are less distinctive, and fail to localize matches reliably (e.g., a patch covering small textured area on one giraffe's body may match anywhere in another giraffe). However, the Dense+SIFT baseline obtains better BBOS for the Applelogo class, likely because its regular shape fits well on a regular patch for some scales. Surprisingly, the shape-based Segment+PHOG baseline does not provide a clear advantage over Dense+SIFT for localization. We suspect this is due to two factors: first, the instability of segmentations across instances, and second, the segments that cover entire objects are not easily matched if there is a viewpoint change or deformation.

Figure 9. Example matches showing BPLR's localization accuracy. Colors in the same row indicate matched regions. Best viewed in color.

Figure 9 illustrates BPLR localization power. For each test image on the left, we select the top five non-overlapping regions based on the foreground matching distance, and display them on the training images to the right. We see matches are consistently localized in spite of scale changes, illumination, and background clutter. Overall, the results in this section indicate that our features' distinctiveness permits reliable localization, a strength for object detection.

4.3. Foreground Discovery with BPLR

Now we examine BPLR's effectiveness for higher-level applications. Our goal in the next experiment is to test whether our approach can improve *foreground discovery*, by replacing the frequently used "superpixels" with BPLRs as base features. In the weakly-supervised foreground (fg) discovery problem [7, 1], the system is given a set of cluttered images that all contain the same object class, and must estimate which pixels are foreground.

Figure 10. Example foreground discovery results using BPLRs. Two examples per class. Ground truth is marked in red. BPLR matching cleanly separates objects from the background in most cases. In some cases, however, we see small leaks near object boundaries (e.g., see the ferry and butterfly), likely due to background regions abutting object boundaries that are confused by strong shape contours.

Approach	Accuracy(%)
BPLR GrabCut (Ours)	**85.6**
Superpixel GrabCut [27]	81.5
Superpixel ClassCut [1]	83.6
Superpixel Spatial Topic Model [7]	67.0

Table 1. Foreground discovery results, compared to several state-of-the-art methods. Using BPLR regions with a GrabCut-based solution, we obtain the best accuracy to date on the Caltech-28 dataset. (See text for details.)

Figure 11. Impact of BPLR matching on fg likelihood. Red areas indicate where fg likelihood exceeds that of bg. The initial fg color model is incorrect (2nd img), but BPLR matches to other bonsai images correctly predict the object location (3rd img). Combining the color model and BPLR matches (4th img), we obtain an accurate fg estimate (last image).

We design a simple model for this task using BPLRs. It is much like the GrabCut [27] baseline defined in [1], in that we initialize a fg color model from the central 25% of the images and a bg color model from the rest, and then solve a standard graph-cut binary labeling problem. However, we replace the superpixel nodes used in [1] with our BPLRs, and add an additional term to the node potential based on the BPLR matches. The new term reflects that we prefer to label BPLR regions as fg if they match well to other BPLRs in images of the same class (the assumption being that same-class backgrounds are uncorrelated). Specifically, let m_f denote the distance from a BPLR's descriptor to its nearest neighbor among the same-class images, and let m_b denote the distance to its nearest neighbor in the images from other classes; if $m_b - m_f$ is positive, we use it to adjust the color-based fg likelihood (see Figure 11). We average likelihoods wherever BPLRs overlap to obtain a single value per pixel. We test with the setup prescribed in previous work [1, 7], which uses 28 Caltech classes, 30 images each, and measures accuracy by the percentage of correctly classified pixels.

Table 1 shows the results. BPLR yields the best accuracy, showing its strength at capturing class-specific shapes in a highly repeatable manner. Our improvement over the GrabCut baseline directly isolates the contribution of BPLR matching (5% gain). Our improvements over the more elaborate models of [7, 1] suggest that even with a simpler labeling objective, BPLRs are preferable to the less-repeatable superpixel base features. Figure 10 shows some example segmentations computed with our method.

4.4. Object Classification with BPLR

Finally, we apply our features to object recognition on the Caltech-101. We again employ a relatively simple classification model on top of the BPLRs, to help isolate their impact. Specifically, we use the Naive Bayes Nearest-Neighbor (NBNN) classifier [5], which sums the NN feature match distances from a test image to those pooled among the training images of each class, and picks the class that produces the lowest matching distance. We follow standard procedures, using 15 random images per class to train and test respectively.

Table 2 compares our results to those using NBNN with alternative feature extractors. With the same PHOG descriptor, our method outperforms the baselines by a large margin. Furthermore, we make a 10% improvement over Dense+SIFT, the strongest baseline; while both extract a similar number of features, our shape-preserving features have a clear advantage over the uniform patch sampling.

Table 3 compares our results to existing single-feature NN-based results reported in the literature. BPLR offers noticeable gains over almost all such methods, even some that use learned metrics [11]. Overall, these results show that our shape-preserving dense features lead to more reliable matches than alternative extraction methods, and coupled with a very simple model are quite effective for object classification.

Feature	Accuracy(%)
BPLR+PHOG (Ours)	**61.1**
Dense+SIFT	55.2
Segment+PHOG	37.6
Dense+PHOG	27.9

Table 2. Direct comparison of BPLR to other feature detectors on the Caltech-101. *The only thing varying per method is the feature extractor*, and our method provides the most accurate results.

Feature	Accuracy(%)
BPLR+PHOG (Ours)	61.1
NBNN+Dense SIFT [5]	65.0
AsymRegionMatch+Geom [13]	61.3
SVM-KNN [32]	59.1
GB+Learned distance [11]	58.4
Segment+Learned distance [11]	55.1
GB+Vote [3]	52
BergMatching [4]	48.0

Table 3. Comparison to existing results on the Caltech-101 that use nearest neighbor-based classifiers. Ours are among the leading results.[4]

5. Conclusions

We introduced a dense local detector that produces repeatable shape-preserving regions via a novel segmentation-driven sampling strategy. As shown through extensive experiments, the key characteristics that distinguish BPLR from existing detectors are: 1) it can improve the ultimate descriptors' distinctiveness, while still retaining thorough coverage of the image, 2) it exploits segments' shape cues without relying on them directly to generate regions, thereby retaining robustness to segmentation variability, and 3) its generic bottom-up extraction makes it applicable whether or not prior class knowledge is available. As such, BPLR can serve as a useful new addition to researchers' arsenal of well-used local feature techniques; to make it easy to do so, we share our code.[5]

Acknowledgements: This research is supported in part by NSF EIA-0303609, the Luce Foundation, LLNL B594497 and a Fellowship from the ILJU Foundation, Korea. Thanks to Marius Muja for the FLANN code and Pablo Arbelaez for the segmentation code.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Classcut for Unsupervised Class Segmentation. In *ECCV*, 2010.

[2] P. Arbelaez, M. Marie, C. Fowlkes, and J. Malik. From Contours to Regions: An Empirical Evaluation. In *CVPR*, 2009.

[3] A. Berg. *Shape Matching and Object Recognition.* PhD thesis, Compute Science Division, Berkeley, 2005.

[4]The authors of [5] report 65.0% when using dense SIFT with NBNN (as shown in Table 3); despite substantial effort, our implementation of this baseline yields only 55.2% (as shown in Table 2). We attribute the discrepancy to some unknown difference in the feature sampling rate or approximate neighbor search procedure parameters.

[5]http://vision.cs.utexas.edu/projects/bplr/bplr.html

[4] A. Berg, T. Berg, and J. Malik. Shape Matching and Object Recognition Low Distortion Correspondences. In *CVPR*, 2005.

[5] O. Boiman, E. Shechtman, and M. Irani. In Defense of Nearest-Neighbor Based Image Classification. In *CVPR*, 2008.

[6] A. Bosch, A. Zisserman, and X.Munoz. Representing Shape with a Spatial Pyramid Kernel. 2007.

[7] L. Cao and L. Fei-Fei. Spatially Coherent Latent Topic Model for Concurrent Segmentation and Classification of Objects and Scenes. In *ICCV*, 2007.

[8] G. Carneiro and A. Jepson. Flexible Spatial Configuration of Local Image Features. *PAMI*, 29(12):2089–2104, 2007.

[9] V. Ferrari, T. Tuytelaars, and L. Gool. Object Detection by Contour Segment Networks. In *ECCV*, 2006.

[10] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie. Weakly Supervised Object Localization with Stable Segmentations. In *ECCV*, 2008.

[11] C. Gu, J. Lim, P. Arbelaez, and J. Malik. Recognition Using Regions. In *CVPR*, 2009.

[12] F. Jurie and C. Schmid. Scale-Invariant Shape Features for Recognition of Object Categories. In *CVPR*, 2004.

[13] J. Kim and K. Grauman. Asymmetric Region-to-Image Matching for Comparing Images with Generic Object Categories. In *CVPR*, 2010.

[14] P. Koniusz and K. Mikolajczyk. Segmentation Based Interest Points and Evaluation of Unsupervised Image Segmentation Methods. In *BMVC*, 2009.

[15] Y. J. Lee and K. Grauman. Foreground Focus: Unsupervised Learning from Partially Matching Images. *IJCV*, 85(2), May 2009.

[16] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Multiscale Symmetric Part Detection and Grouping. In *ICCV*, 2009.

[17] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using Contours to Detect and Localize Junctions in Natural Images. In *CVPR*, 2008.

[18] T. Malisiewicz and A. Efros. Improving Spatial Support for Objects via Multiple Segmentations. In *BMVC*, 2007.

[19] J. Matas, O. Chum, M. Urba, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *BMVC*, 2002.

[20] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *IJCV*, 1(60):63–86, October 2004.

[21] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A Comparison of Affine Region Detectors. *IJCV*, 65:43–72, 2005.

[22] M. Muja and D. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In *VISAPP*, 2009.

[23] E. Nowak, F. Jurie, and B. Triggs. Sampling Strategies for Bag-of-Features Image Classification. In *ECCV*, 2006.

[24] C. Pantofaru, G. Dorko, C. Schmid, and M. Hebert. Combining Regions and Patches for Object Class Localization. In *Beyond Patches, Workshop in conjunction with CVPR*, 2006.

[25] T. Quack, V. Ferrari, B. Leibe, and L. Gool. Efficient Mining of Frequent and Distinctive Feature Configurations. In *ICCV*, 2007.

[26] X. Ren and J. Malik. Learning a Classification Model for Segmentation. In *ICCV*, 2003.

[27] C. Rother, V. Komogorov, and A. Blake. Grabcut: Interactive Foreground Extraction Using Iterated Graph Cuts. *SIGGRAPH*, 23:309–314, 2004.

[28] M. Sabuncu and P. Ramadge. Using Spanning Graphs for Efficient Image Registration. *IEEE Trans. on Image Processing*, 17, 2008.

[29] T. Sebastian, P. Klein, and B. Kimia. Recognition of Shapes by Editing their Shock Graphs. *PAMI*, 26:551–571, 2004.

[30] S. Todorovic and N. Ahuja. Learning Subcategory Relevances for Category Recognition. In *CVPR*, 2008.

[31] T. Tuytelaars. Dense Interest Points. In *CVPR*, 2010.

[32] H. Zhang, A. Berg, M. Marie, and J. Malik. SVM-KNN: Discriminative Nearest Neighbor Classfication for Visual Category Recognition. In *CVPR*, 2006.